



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



4EFCFC6D-652D-4AB2-92A6-7BD4FB013539

SPRING2014_CSCI570_EXAM2

#1 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

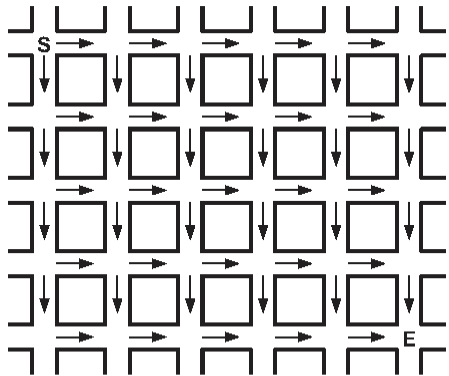


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



37E5999F-73F6-4CEF-95FB-15C979EE944C

SPRING2014_CSCI570_EXAM2

#1 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

5C0CC393-B52F-4CBA-8457-3B4C732E6622

SPRING2014_CSCI570_EXAM2

#1 9 of 10



Additional Space



03F2F3A4-B0A0-4F61-BF8F-DCF01402C03F

SPRING2014_CSCI570_EXAM2

#1 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



46971CE1-5212-4F86-82EB-E5EFAADEC799

SPRING2014_CSCI570_EXAM2

#2 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



EE1B7CE6-30EB-493B-BD94-EBF3F77D3E24

SPRING2014_CSCI570_EXAM2

#2 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

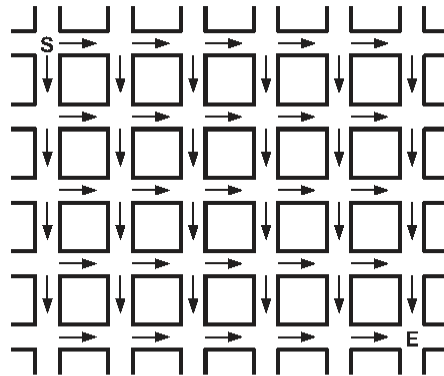


Figure A.

-
- The diagram shows a 2D lattice of squares. A central horizontal path of three squares is missing, creating a defect. The path is labeled 'S' at the top and 'E' at the bottom. Arrows indicate a flow from left to right and top to bottom.

Figure B.



0DFF66D3-4F02-4D02-ADDA-CCBDFECBA52B

SPRING2014_CSCI570_EXAM2

#2 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

04BA491B-9B38-4A8A-8CFD-9143B514B559

SPRING2014_CSCI570_EXAM2

#2 9 of 10



Additional Space



F204BFBE-775D-42EF-8728-1DF2674D272B

SPRING2014_CSCI570_EXAM2

#2 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



32F4FCDB-0856-4A75-AC7B-7431BAF13F1A

SPRING2014_CSCI570_EXAM2

#3 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

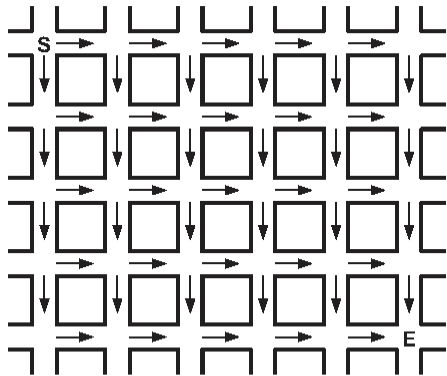


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



92157967-D021-4F4F-9024-49DB669CC6F3

SPRING2014_CSCI570_EXAM2

#3 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

6F0B1C68-793A-4D51-9A7A-3B665044D094

SPRING2014_CSCI570_EXAM2

#3 9 of 10



Additional Space



55D8A806-4FDE-4EF2-9764-9DD8B158F030

SPRING2014_CSCI570_EXAM2

#3 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



2BE83477-24C7-4D0C-9C16-2E7ED1988016

SPRING2014_CSCI570_EXAM2

#4 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



C035BA53-6B71-4CA3-965D-6BA928C1E25B

SPRING2014_CSCI570_EXAM2

#4 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



95E1425D-8D2B-4AE3-81DD-FADE20AE5C1A

SPRING2014_CSCI570_EXAM2

#4 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

2286CA54-9C2F-4710-8295-EDEB3BCCF098

SPRING2014_CSCI570_EXAM2

#4 9 of 10



Additional Space



82FFDC7C-4C92-4350-A689-4021179D992D

SPRING2014_CSCI570_EXAM2

#4 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



36301CC7-0F29-46AB-BDB2-54D7798E5D14

SPRING2014_CSCI570_EXAM2

#5 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

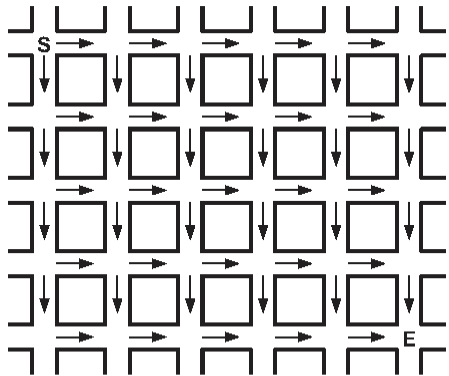


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

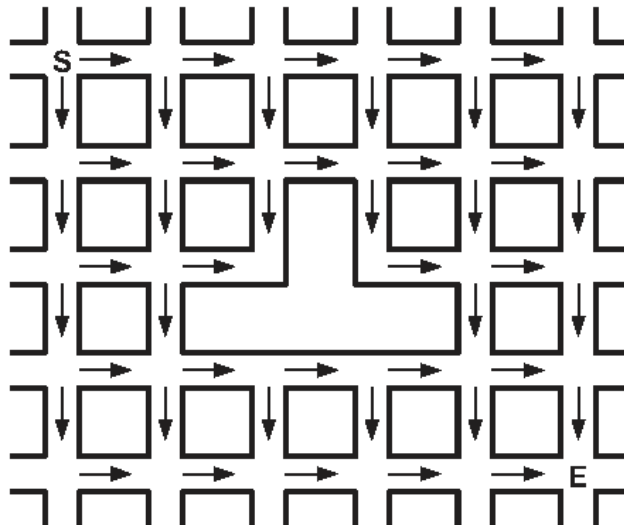


Figure B.



9B2AEDCC-FCCD-43CA-9A17-598D3CC67BFA

SPRING2014_CSCI570_EXAM2

#5 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

A2B5D8C6-CFC4-470E-A048-4527A49DA4D0

SPRING2014_CSCI570_EXAM2

#5 9 of 10



Additional Space



9E9F5BBC-2DA2-46A1-A510-3B3868C4DFBD

SPRING2014_CSCI570_EXAM2

#5 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



BEE41766-B719-4505-BBDA-9D43998DD099

SPRING2014_CSCI570_EXAM2

#6 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

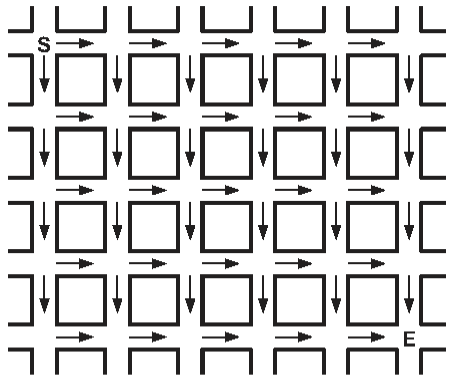


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



45181AA2-A8DD-4642-A41E-E01E0DD5E369

SPRING2014_CSCI570_EXAM2

#6 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

03EA5780-4409-4E87-A048-02DDA8B55BCC

SPRING2014_CSCI570_EXAM2

#6 9 of 10



Additional Space



261E63C7-7EFE-44BB-A166-6026BBC98C5A

SPRING2014_CSCI570_EXAM2

#6 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



D89E012F-C185-4BCF-AC29-B82818742745

SPRING2014_CSCI570_EXAM2

#7 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



0EB2B460-4FCE-451B-8643-3467C932AAA5

SPRING2014_CSCI570_EXAM2

#7 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

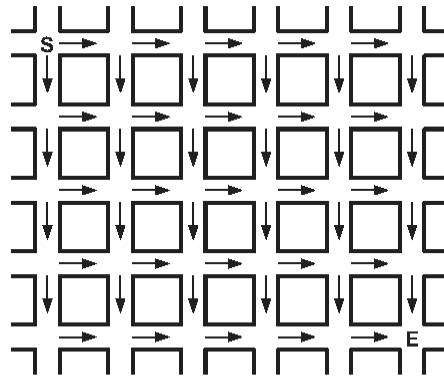


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

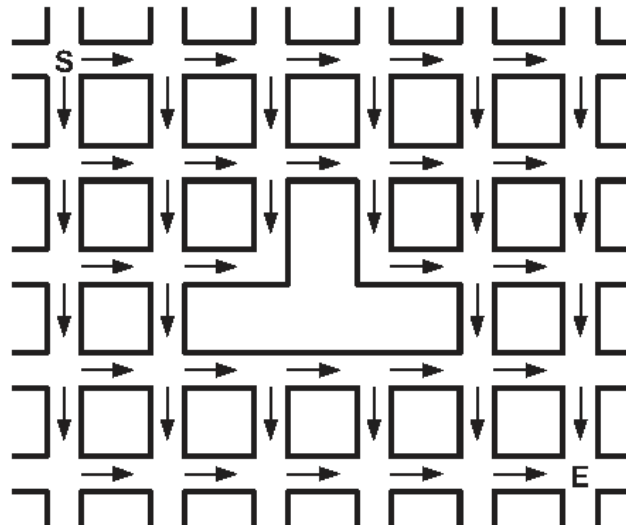


Figure B.



A9B45C25-6D0B-4088-AB41-74FC545E7503

SPRING2014_CSCI570_EXAM2

#7 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

39D6E4FD-03FF-40C1-A45F-9A748573FF1C

SPRING2014_CSCI570_EXAM2

#7 9 of 10



Additional Space



9B24B862-C90E-44CE-8C7E-221B3B6ECF82

SPRING2014_CSCI570_EXAM2

#7 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



A03A9BDD-5E42-4913-8340-D3AA20EF827E

SPRING2014_CSCI570_EXAM2

#8 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



D2F31E84-E973-4E86-8FD9-B1F92F3AD52F

SPRING2014_CSCI570_EXAM2

#8 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

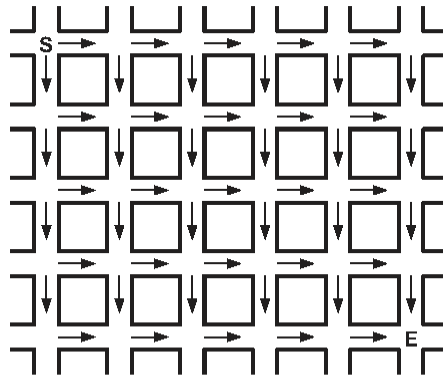


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



43A9198D-7EFC-42D9-B0C8-AE04DEE97495

SPRING2014_CSCI570_EXAM2

#8 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

99B0F19C-6FB2-4F1D-8262-E3537598AD79

SPRING2014_CSCI570_EXAM2

#8 9 of 10



Additional Space



4DB5CA19-0F0B-4558-9743-DEFF6AF084F9

SPRING2014_CSCI570_EXAM2

#8 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



5704FB93-4A2E-4723-98D8-E8E1BA321952

SPRING2014_CSCI570_EXAM2

#9 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



83ECEBE6-292D-4934-A016-A74992DDA539

SPRING2014_CSCI570_EXAM2

#9 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

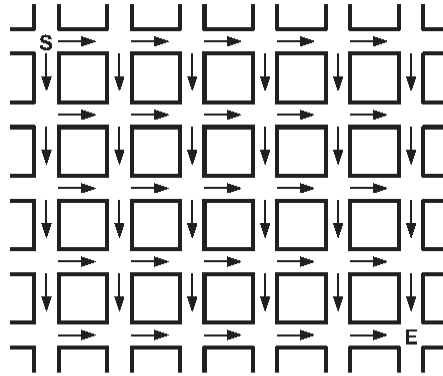


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



8EA7F842-73A6-4C1D-B4E3-B046D8F88C09

SPRING2014_CSCI570_EXAM2

#9 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

2B10128B-5351-4DAF-B77B-C673E7F44829

SPRING2014_CSCI570_EXAM2

#9 9 of 10



Additional Space



3660BFC2-B204-44CC-A69B-1E528BB12BBD

SPRING2014_CSCI570_EXAM2

#9 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



51EBBEF5-644A-4751-8AD4-BB0E221CF646

SPRING2014_CSCI570_EXAM2

#10 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



A89FF30B-B5BB-49C3-A5DE-EBB6853C6F1B

SPRING2014_CSCI570_EXAM2

#10 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



CF7A8DA1-038B-4770-83EB-A699F3C1B2CF

SPRING2014_CSCI570_EXAM2

#10

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

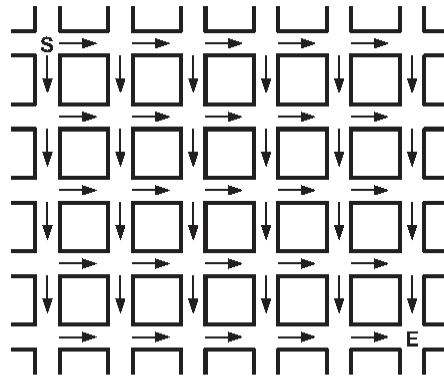


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



89C9BA7A-7063-40DA-87B0-F6C929FDD92A

SPRING2014_CSCI570_EXAM2

#10 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

E5130F39-6F34-43E5-8C4A-9421D1277689

SPRING2014_CSCI570_EXAM2

#10 9 of 10



Additional Space



6A789D85-9D05-4C28-A3B7-5A424C0E4BDE

SPRING2014_CSCI570_EXAM2

#10 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



B25E47A4-51A6-4D0B-8563-6D293CD17524

SPRING2014_CSCI570_EXAM2

#11 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



4E3A1049-4CFE-4396-989C-479457808ABC

SPRING2014_CSCI570_EXAM2

#11

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

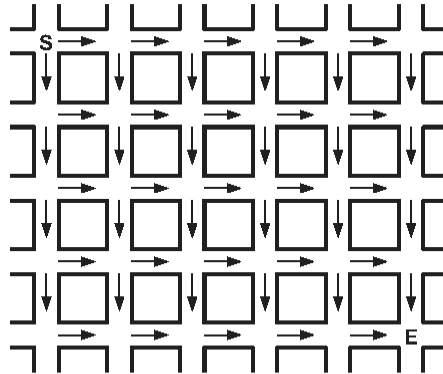


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\cdot, \cdot)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\cdot, \cdot)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



8268A8B2-CC79-440E-99A3-09DE4092B57D

SPRING2014_CSCI570_EXAM2

#11 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

0D65D681-819A-440D-917B-08E84175EE07

SPRING2014_CSCI570_EXAM2

#11

9 of 10



Additional Space



783D052A-CCD4-4072-BE02-E2D7FE363438

SPRING2014_CSCI570_EXAM2

#11 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



5CDA0A3F-72FB-4AF1-8146-83200088630E

SPRING2014_CSCI570_EXAM2

#12 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



6713EE9B-B4C2-4075-884A-85A0184D1D8D

SPRING2014_CSCI570_EXAM2

#12 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

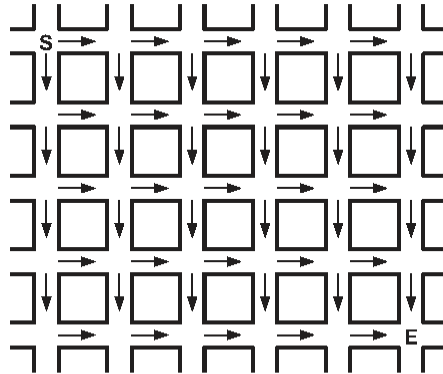


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



B6128382-D1CE-4A3D-8FBC-695134DF6BB5

SPRING2014_CSCI570_EXAM2

#12 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

3EBD8E9A-579B-4D4B-BABD-EFC7829EC9EA

SPRING2014_CSCI570_EXAM2

#12 9 of 10



Additional Space



59C408B2-0002-484D-AF53-5EF03F4855C5

SPRING2014_CSCI570_EXAM2

#12 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



1772B822-08C6-48CF-A498-196BF95B0DF9

SPRING2014_CSCI570_EXAM2

#13 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

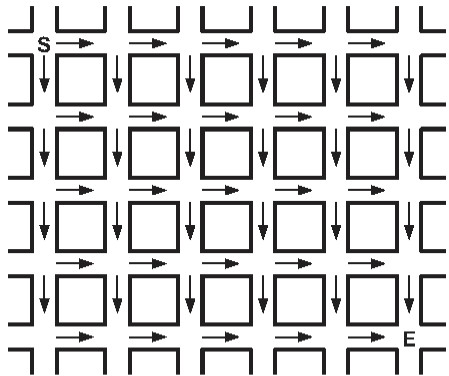


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



42AA0351-4EDA-4F1D-B7F5-4AD007B770B0

SPRING2014_CSCI570_EXAM2

#13 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

CAD9B116-62C9-4048-B217-9434365183D9

SPRING2014_CSCI570_EXAM2

#13 9 of 10



Additional Space



5BF920A2-0910-4C90-BEB3-3713EE23CF3F

SPRING2014_CSCI570_EXAM2

#13 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



5C57F3B6-1A9D-4DBD-AB2A-B13ED3852A2A

SPRING2014_CSCI570_EXAM2

#14 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

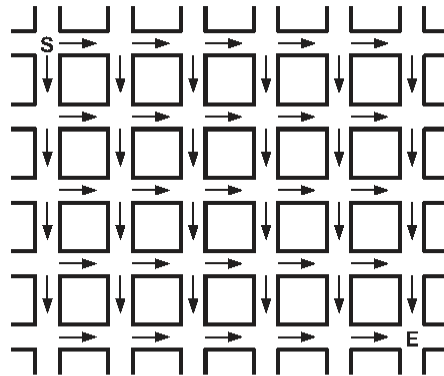


Figure A.

-
- The diagram shows a 5x5 grid of squares. Arrows on the edges of the squares indicate a flow from the top-left towards the bottom-right. A path of squares is highlighted in black, starting from the top-left square and ending at the bottom-right square. The path is labeled 'S' at the top-left and 'E' at the bottom-right.

Figure B.



592C0DC8-9C24-4D30-B16F-4787C99CB7F7

SPRING2014_CSCI570_EXAM2

#14 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

F0DC7ADB-1EFF-41CF-896B-028B14A73CD8

SPRING2014_CSCI570_EXAM2

#14

9 of 10



Additional Space



F2FAA7E5-9568-46E3-B407-DE880E16C909

SPRING2014_CSCI570_EXAM2

#14 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



557119CE-B224-46B8-B737-828B8E74E70B

SPRING2014_CSCI570_EXAM2

#15 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



254807AC-C596-484F-A20A-222FAC40BEBE

SPRING2014_CSCI570_EXAM2

#15

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

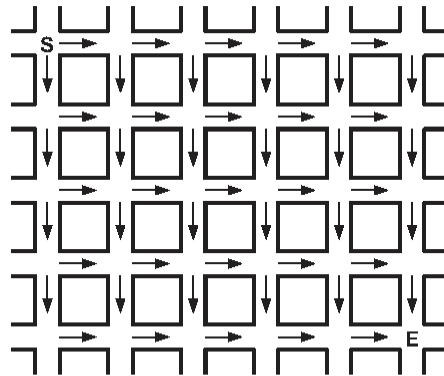


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

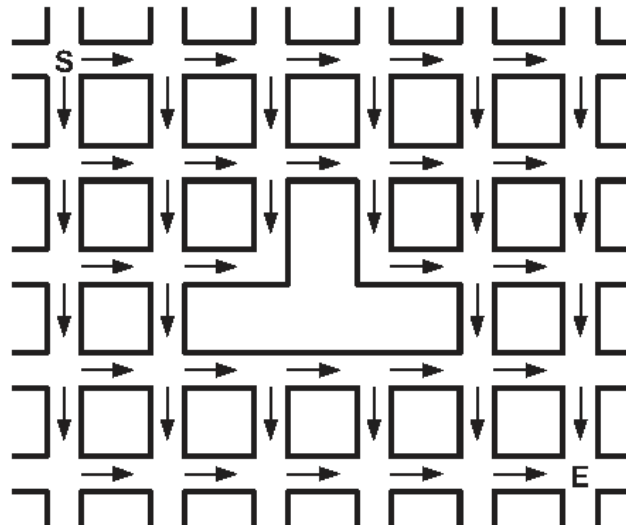


Figure B.



F3FEBDA3-821B-4342-A4C3-50AFA9AC2965

SPRING2014_CSCI570_EXAM2

#15 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

C203B623-D5BF-46BA-A3E9-F070037849FD

SPRING2014_CSCI570_EXAM2

#15 9 of 10



Additional Space



64FBBEDF-FFEA-497C-A0CB-8784DC104802

SPRING2014_CSCI570_EXAM2

#15 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



E1AEE011-E0FB-4A05-BAD9-7BE3236CD80E

SPRING2014_CSCI570_EXAM2

#16 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

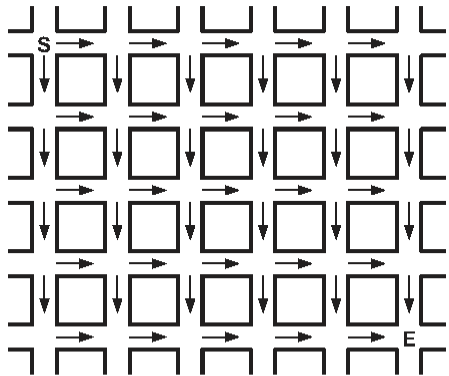


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



F442582A-36A3-49A6-B471-1254A5053CD5

SPRING2014_CSCI570_EXAM2

#16 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

34589D72-C965-4D66-A782-56AF0BB08FC3

SPRING2014_CSCI570_EXAM2

#16 9 of 10



Additional Space



F02854D1-387C-4B69-B0A1-54F3D5909978

SPRING2014_CSCI570_EXAM2

#16 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



3CB8E2F4-A28D-410A-9E67-655915AB1443

SPRING2014_CSCI570_EXAM2

#17 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



DA029767-D693-4C5F-B70E-6A94FBCA7AD9

SPRING2014_CSCI570_EXAM2

#17 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



737E97B8-5BD2-4635-B98A-499D304A98EE

SPRING2014_CSCI570_EXAM2

#17 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

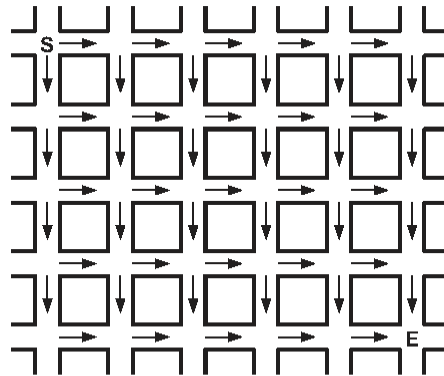


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



DD1A0D08-B669-44C5-89CA-7B93217C145A

SPRING2014_CSCI570_EXAM2

#17 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

759F47B1-0C55-4FFA-B496-7A495F20AB25

SPRING2014_CSCI570_EXAM2

#17 9 of 10



Additional Space



79BDDD23-E5DE-45CF-85FA-284DCF8A0A09

SPRING2014_CSCI570_EXAM2

#17 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



E37C4E44-5AB8-4681-857C-8A4B75F6650A

SPRING2014_CSCI570_EXAM2

#18 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



23D029E2-B35B-4EA1-AF1F-0BF1737787AB

SPRING2014_CSCI570_EXAM2

#18 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

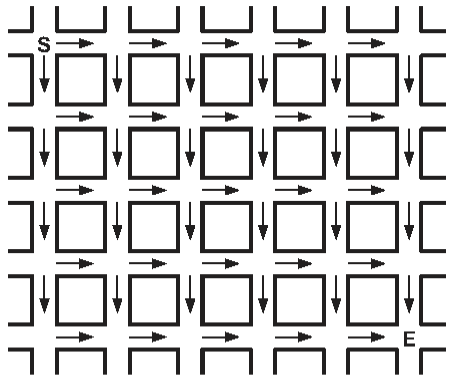


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



0E84DC93-1464-4EF4-9E43-B2E3911FAE27

SPRING2014_CSCI570_EXAM2

#18 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

6BAE27A9-51F8-4674-9A2B-17F29218660D

SPRING2014_CSCI570_EXAM2

#18 9 of 10



Additional Space



B4FF074D-8806-40BD-8C4A-E4537CB87E14

SPRING2014_CSCI570_EXAM2

#18 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



5695CFFB-B608-4E3E-84D3-9EEFC125F3C8

SPRING2014_CSCI570_EXAM2

#19 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

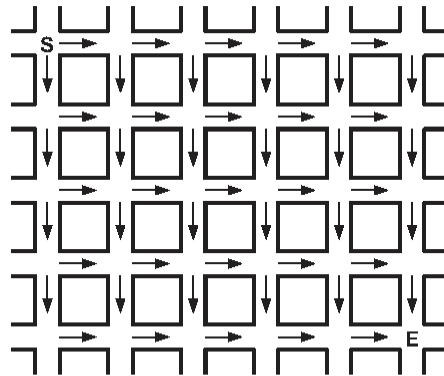


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



56CB7830-FEFA-406C-A3A5-67EF89DF24F9

SPRING2014_CSCI570_EXAM2

#19 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

92E28574-E5CD-4DAF-BD07-3898D1DAE46C

SPRING2014_CSCI570_EXAM2

#19 9 of 10



Additional Space



1A26A25F-9057-4408-AE01-8FE7AC96DEDA

SPRING2014_CSCI570_EXAM2

#19 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



3AD869CB-04B0-43C2-A7FD-11D3D6151E91

SPRING2014_CSCI570_EXAM2

#20 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



4DCEE7D2-A923-4E34-A2AC-1C7CF5BC9FBE

SPRING2014_CSCI570_EXAM2

#20

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



3C105FEB-5E2A-4E25-BC4A-23AF76610B4F

SPRING2014_CSCI570_EXAM2

#20

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

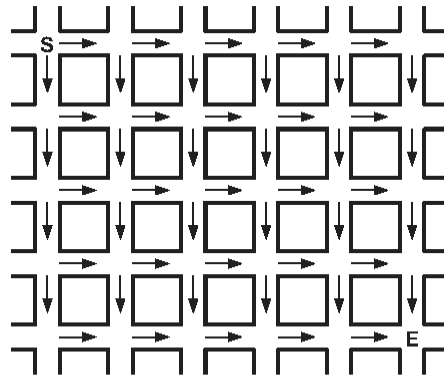


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

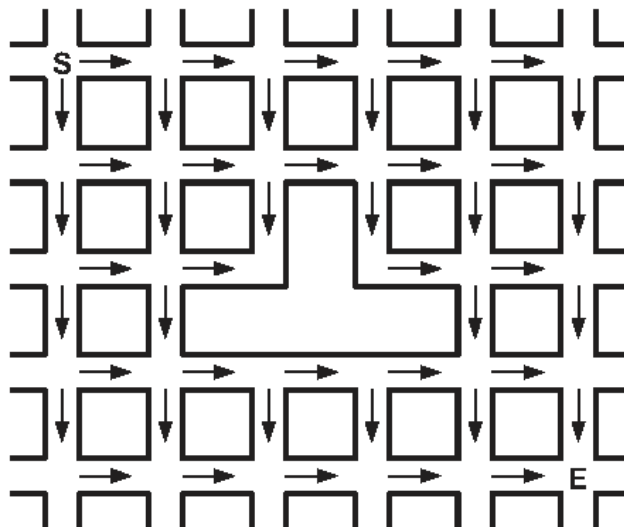


Figure B.



F6F5622E-B948-4DBB-B622-3B02E25F1927

SPRING2014_CSCI570_EXAM2

#20 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

E7CAF3CE-4D06-4A63-8655-08A84B25DBD9

SPRING2014_CSCI570_EXAM2

#20 9 of 10



Additional Space



154BB3FC-6F9C-4753-B797-5BB745208830

SPRING2014_CSCI570_EXAM2

#20 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



30CFC3B2-73E2-4EC8-A196-E2A410A6F950

SPRING2014_CSCI570_EXAM2

#21 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



2C343733-B846-4F7E-AB4B-1179EEF0D31C

SPRING2014_CSCI570_EXAM2

#21 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

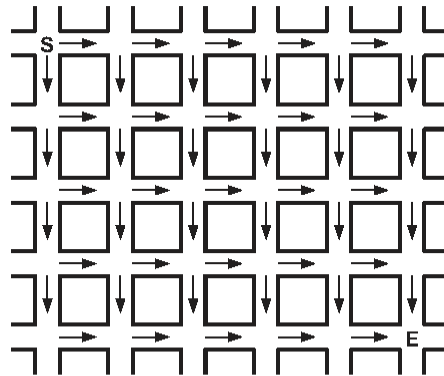


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



97624E43-1FEE-45FB-AE67-4C2178C6E0DE

SPRING2014_CSCI570_EXAM2

#21 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

D94F2F4D-9851-4117-B4F1-A9961294ACBC

SPRING2014_CSCI570_EXAM2

#21 9 of 10



Additional Space



71FA0FFE-7BDB-47ED-BA13-0C236E963F21

SPRING2014_CSCI570_EXAM2

#21 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



68ED9606-8D3D-4A89-B249-BB90D9426D72

SPRING2014_CSCI570_EXAM2

#22 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



EDBA76AE-9F2D-462A-9123-5CC5DE13EAF3

SPRING2014_CSCI570_EXAM2

#22 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

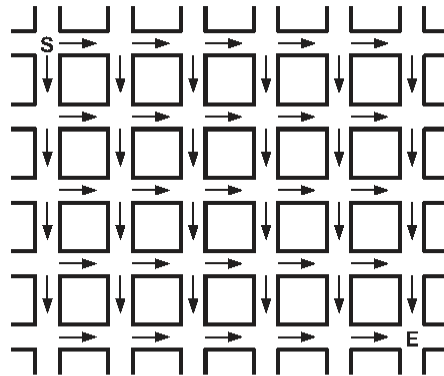


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



BA9F5889-4B94-4018-90F2-EEC9AD9B2E29

SPRING2014_CSCI570_EXAM2

#22 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

0403CBBB-6821-4ABA-A481-2E7D8C9F3674

SPRING2014_CSCI570_EXAM2

#22 9 of 10



Additional Space



4109D4A9-91B0-474D-A421-2E0C98830301

SPRING2014_CSCI570_EXAM2

#22 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



7DE3F97F-035D-4023-8B06-242C7DDBE959

SPRING2014_CSCI570_EXAM2

#23 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

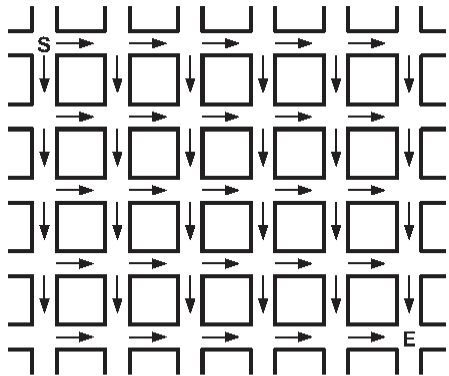


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

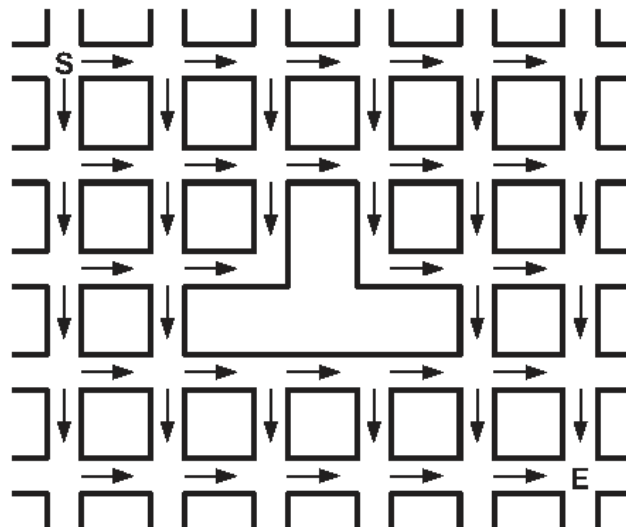


Figure B.



6B1B096E-9084-4716-9359-3A4244BEA392

SPRING2014_CSCI570_EXAM2

#23 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

4EC6F18C-0A72-4FE7-8EFD-804005C334CB

SPRING2014_CSCI570_EXAM2

#23 9 of 10



Additional Space



7426D439-2DC9-4743-BA60-CCFADBD27618

SPRING2014_CSCI570_EXAM2

#23 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



128878BE-BBA5-404C-80AF-FFD08CD3AB81

SPRING2014_CSCI570_EXAM2

#24 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



8468D77C-1C28-4E73-944E-0DF757248973

SPRING2014_CSCI570_EXAM2

#24

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



55BE5792-9FF3-45F9-B769-56529EFC99AE

SPRING2014_CSCI570_EXAM2

#24

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

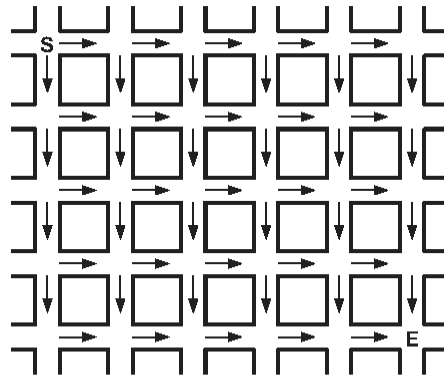


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



1AE6AA6D-EFD9-4355-8CFE-06E988D5ECAE

SPRING2014_CSCI570_EXAM2

#24 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9BE1AA57-8305-4094-814A-C0EEC6E5DEF9

SPRING2014_CSCI570_EXAM2

#24

9 of 10



Additional Space



110CD569-A64B-4072-A2BD-81F0B627DFCC

SPRING2014_CSCI570_EXAM2

#24 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



023865A3-6EF0-4FD2-AC82-5C2A82B0211A

SPRING2014_CSCI570_EXAM2

#25 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

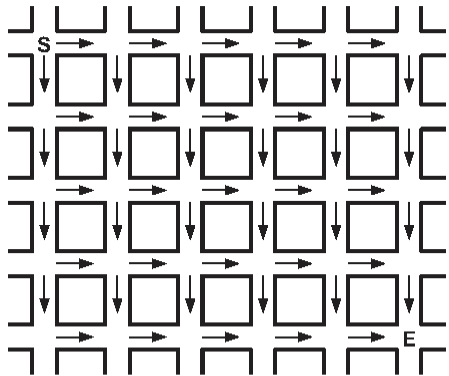


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



2213964A-D0F8-451C-AAE6-FDF8BCEE5F3

SPRING2014_CSCI570_EXAM2

#25 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

EA1B8D9C-DECA-4541-BD2A-F982895D395B

SPRING2014_CSCI570_EXAM2

#25 9 of 10



Additional Space



97047F5B-ED39-4BBC-B093-7D0DBAC373CF

SPRING2014_CSCI570_EXAM2

#25 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



5B19CC1E-E7CA-4DDB-8253-EFB2CA147306

SPRING2014_CSCI570_EXAM2

#26 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



F96A7F68-6153-48A2-98E8-918910BB8A1D

SPRING2014_CSCI570_EXAM2

#26

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

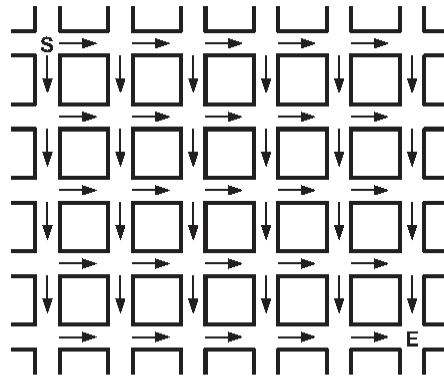


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



23928CCB-C996-438E-89AC-B10AE507DE14

SPRING2014_CSCI570_EXAM2

#26 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

5DF37A0F-E6F9-4156-AF02-CA5094710EA6

SPRING2014_CSCI570_EXAM2

#26 9 of 10



Additional Space



6AC84322-DBCD-47F3-BA25-78D69855F9D5

SPRING2014_CSCI570_EXAM2

#26 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



ECCB7BEB-0D72-41E9-ACB5-6149409D5C71

SPRING2014_CSCI570_EXAM2

#27 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

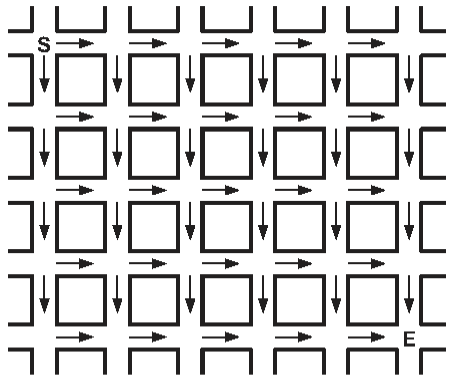


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



E7B53046-76F0-440C-B1F0-4DC065A744E6

SPRING2014_CSCI570_EXAM2

#27 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

F1B97393-CBCF-4400-9925-958085A7A9A0

SPRING2014_CSCI570_EXAM2

#27

9 of 10



Additional Space



F88B34DF-9023-470C-AD4B-1BA543A63AD2

SPRING2014_CSCI570_EXAM2

#27 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



3E58C408-72DD-481F-A503-4E1F8B1667B9

SPRING2014_CSCI570_EXAM2

#28

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



8E2F5B5D-670D-4417-8195-49846038BAEA

SPRING2014_CSCI570_EXAM2

#28

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

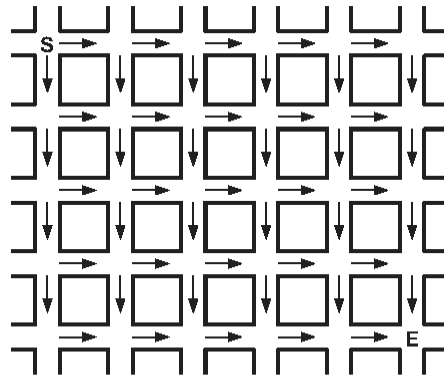


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



93C24106-093F-40A5-B642-15F19B757F82

SPRING2014_CSCI570_EXAM2

#28 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

7EB22238-BD36-4CC5-B991-6BFAD44529C5

SPRING2014_CSCI570_EXAM2

#28 9 of 10



Additional Space



D20CD70E-EC21-4FD4-8C16-3140422A313F

SPRING2014_CSCI570_EXAM2

#28 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



6A608AA5-FBFB-46DA-890E-AE16FC9865CF

SPRING2014_CSCI570_EXAM2

#29 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

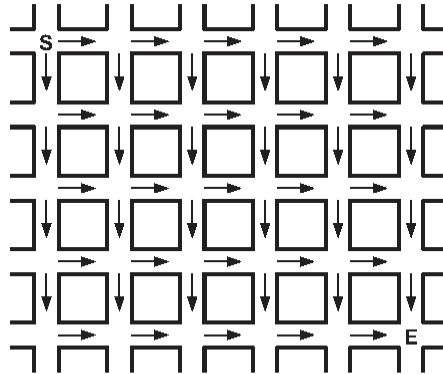


Figure A.



1FA5211E-6D32-4C9D-A733-F24D9094B817

SPRING2014_CSCI570_EXAM2

#29 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

37DB34CC-A748-4E2D-BDEA-F4EF16FF75E0

SPRING2014_CSCI570_EXAM2

#29 9 of 10



Additional Space



9CA08533-4BDD-437E-8F48-0D253966EC5C

SPRING2014_CSCI570_EXAM2

#29

10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



5E8BB1DA-F530-47AA-A0AF-9F13E60B23CF

SPRING2014_CSCI570_EXAM2

#30 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



2FEB8F32-6041-4C10-8362-F512A0134232

SPRING2014_CSCI570_EXAM2

#30 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



AB88C744-81CF-4D55-BAAC-3E61A9039940

SPRING2014_CSCI570_EXAM2

#30 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

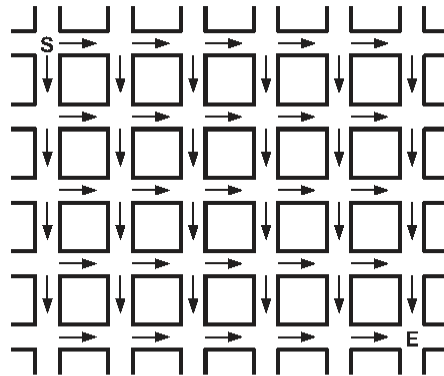


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



1A5EDF82-3D5A-4CFA-BC32-2E969917DA27

SPRING2014_CSCI570_EXAM2

#30 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

21065194-4724-45C8-BDF7-473633632C7F

SPRING2014_CSCI570_EXAM2

#30 9 of 10



Additional Space



F0FF23AC-AFFF-42CF-9BB1-A01C3BA1A5B7

SPRING2014_CSCI570_EXAM2

#30 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



9762F940-8040-4465-8D88-D8BCF7FC3535

SPRING2014_CSCI570_EXAM2

#31 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

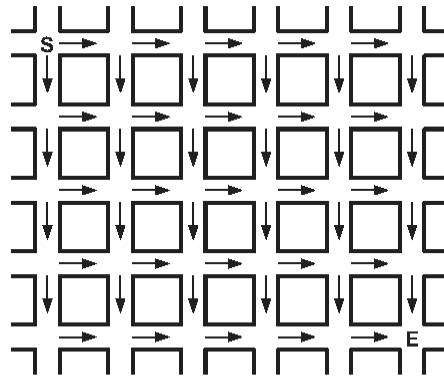


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\cdot, \cdot)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\cdot, \cdot)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



8C1230EB-5DA6-44CF-996C-23E2932ABB6B

SPRING2014_CSCI570_EXAM2

#31 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

7FB9CB60-B337-40F7-AD7E-0F06289CB7AA

SPRING2014_CSCI570_EXAM2

#31 9 of 10



Additional Space



DDAC3AC8-A647-403B-9F97-08C97C5FA446

SPRING2014_CSCI570_EXAM2

#31 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



A1E1464C-C829-4A31-A251-10B9A805D97C

SPRING2014_CSCI570_EXAM2

#32 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

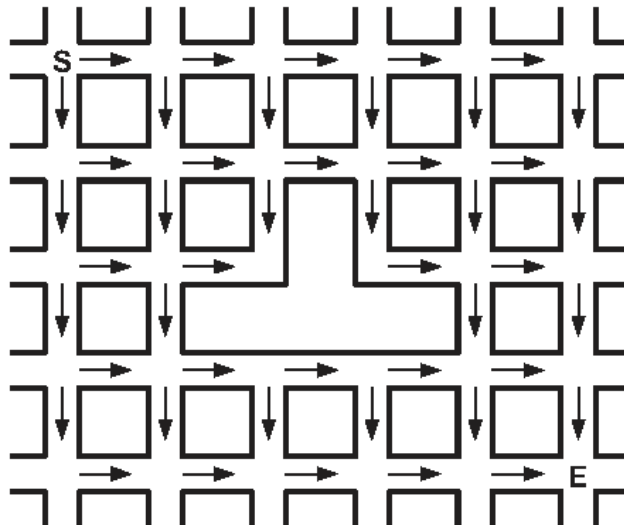


Figure B.



F52437E8-7F03-49F1-BC02-BF64966DE44F

SPRING2014_CSCI570_EXAM2

#32 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

372EC19D-C5FF-4233-9738-375E53320014

SPRING2014_CSCI570_EXAM2

#32 9 of 10



Additional Space



8B658C59-5E46-4076-8696-31DD0A1E1E84

SPRING2014_CSCI570_EXAM2

#32 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



D88A359F-E5B5-468D-A2AA-402C46A4FD19

SPRING2014_CSCI570_EXAM2

#33 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



7DAC4129-6961-402E-95F8-81827B05F92A

SPRING2014_CSCI570_EXAM2

#33 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

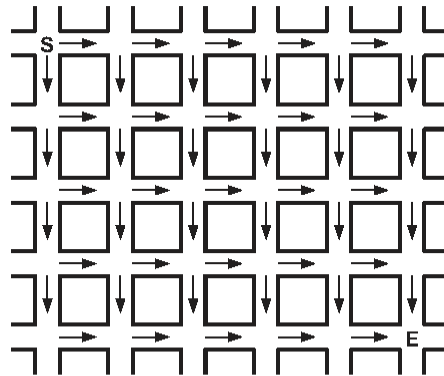


Figure A.



46A3711F-FD5B-4191-9B2F-C6E2A234139B

SPRING2014_CSCI570_EXAM2

#33 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

2C7C698E-16B6-4ACC-A04F-DD1B17FEE440

SPRING2014_CSCI570_EXAM2

#33

9 of 10



Additional Space



2B39CAFC-F801-49F2-A1E6-B049F5365646

SPRING2014_CSCI570_EXAM2

#33 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



1428240B-A0DD-48D3-B5F4-CF52FB7E3477

SPRING2014_CSCI570_EXAM2

#34 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

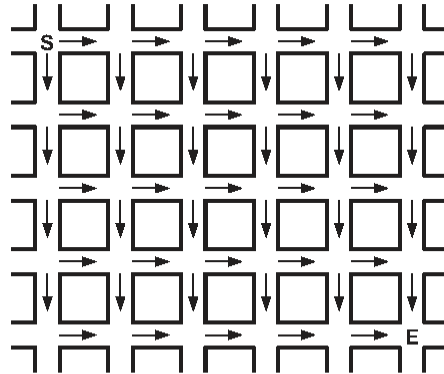


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



23ECC538-0283-4D07-B014-C068BA621DFF

SPRING2014_CSCI570_EXAM2

#34 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

C6AD51D1-51F0-4010-A43B-6E690161396E

SPRING2014_CSCI570_EXAM2

#34 9 of 10



Additional Space



239E75DD-BEB2-444C-84CC-C552254B0E0F

SPRING2014_CSCI570_EXAM2

#34 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



91425CCF-AA58-4942-A5E0-B8917E7B7F8E

SPRING2014_CSCI570_EXAM2

#35 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



5FD7FEBC-1E69-4A68-B6C8-07E1507BC166

SPRING2014_CSCI570_EXAM2

#35

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

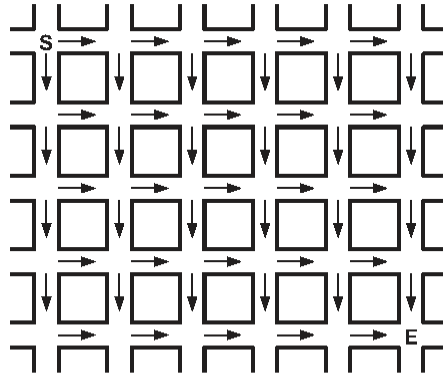


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



CFBF7A82-49BD-4A93-A47C-A9333652014C

SPRING2014_CSCI570_EXAM2

#35 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

4514579F-C077-4F67-84E7-F0274BBB9B9E

SPRING2014_CSCI570_EXAM2

#35 9 of 10



Additional Space



1F703B38-FDFE-4F2B-9CE1-1396EDC53F49

SPRING2014_CSCI570_EXAM2

#35 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



F5C29DCC-7959-405F-9B2E-7ECFC7598E79

SPRING2014_CSCI570_EXAM2

#36 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



11ECB9FB-A235-4A61-9FB2-7299948478F8

SPRING2014_CSCI570_EXAM2

#36

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

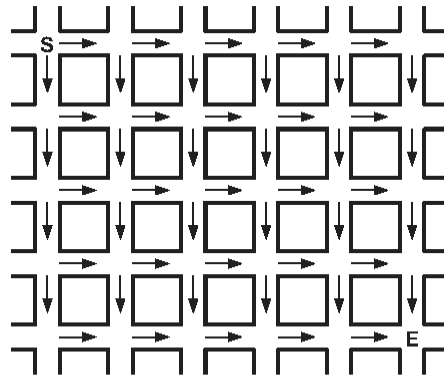


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



6E083605-EEF9-4786-9564-1242E9F88636

SPRING2014_CSCI570_EXAM2

#36 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

02B23C5A-4F65-4D52-91B4-4817922DB8C4

SPRING2014_CSCI570_EXAM2

#36 9 of 10



Additional Space



A39BEE65-D8EB-48C4-B06E-AD9115F5E812

SPRING2014_CSCI570_EXAM2

#36 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



9F8EED8D-4E52-453B-ACAC-22E2805B4AA1

SPRING2014_CSCI570_EXAM2

#37 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



0EFF87C4-80F3-42CF-A7D2-3306C690021D

SPRING2014_CSCI570_EXAM2

#37 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

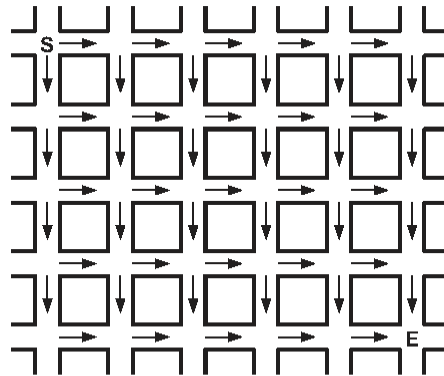


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



78C0904D-5814-4210-8328-4B2E9D2B927B

SPRING2014_CSCI570_EXAM2

#37 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

8C7F7EA5-A4B9-4D1E-B211-C38DDF05655D

SPRING2014_CSCI570_EXAM2

#37 9 of 10



Additional Space



19C8DBBC-90C8-47D7-824A-DB1C6E065E8F

SPRING2014_CSCI570_EXAM2

#37 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



9A49D6CF-D3C2-444C-94C5-DC478CD15673

SPRING2014_CSCI570_EXAM2

#38 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

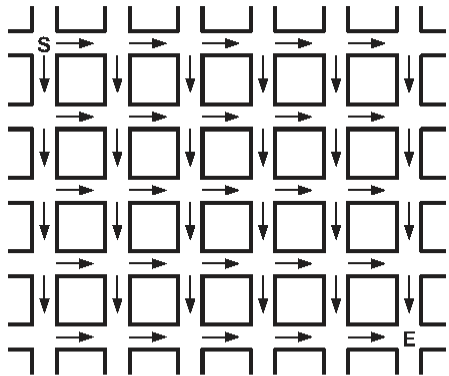


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



37205B13-18CC-47F3-8D97-95D4E02D53C9

SPRING2014_CSCI570_EXAM2

#38 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

BD1D351C-5827-4EB4-8A8A-C04B34B852EB

SPRING2014_CSCI570_EXAM2

#38

9 of 10



Additional Space



6FE45C27-E0C9-4EC0-AEED-62B71E100285

SPRING2014_CSCI570_EXAM2

#38 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



C0B0C926-1909-4E6B-96B0-C814A64DC659

SPRING2014_CSCI570_EXAM2

#39 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

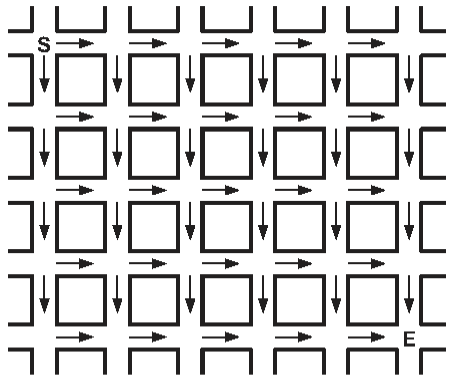


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

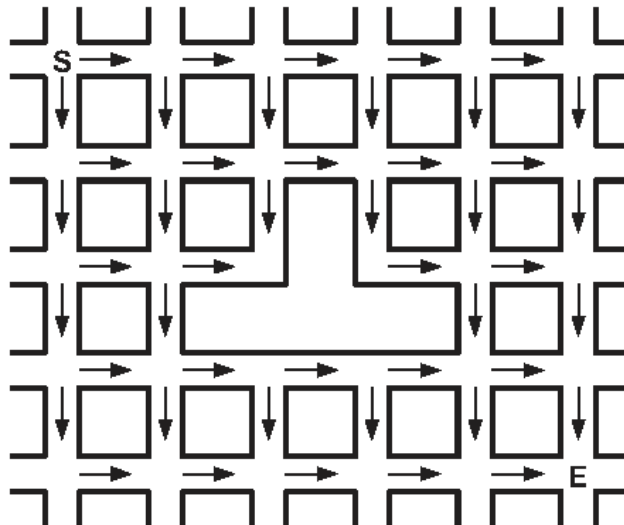


Figure B.



4DAF79B1-C8EA-4573-AA42-A75E984E7D9F

SPRING2014_CSCI570_EXAM2

#39 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

54839D1F-AE2B-4902-92CC-06A38AFAD806

SPRING2014_CSCI570_EXAM2

#39 9 of 10



Additional Space



C4ED77C7-3A42-4F67-BD43-F6D551EE54FD

SPRING2014_CSCI570_EXAM2

#39 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



7D5DAA1E-326D-4C0D-91BE-62B163CBFE21

SPRING2014_CSCI570_EXAM2

#40

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



98C138D4-26BE-4D46-9472-B90F2733EE3F

SPRING2014_CSCI570_EXAM2

#40

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

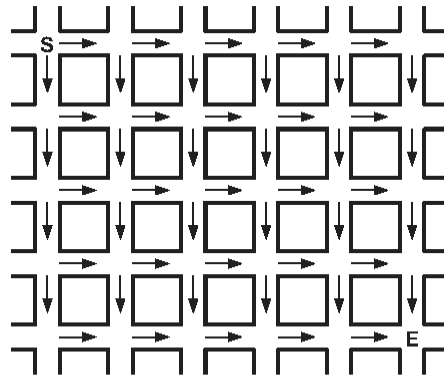


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

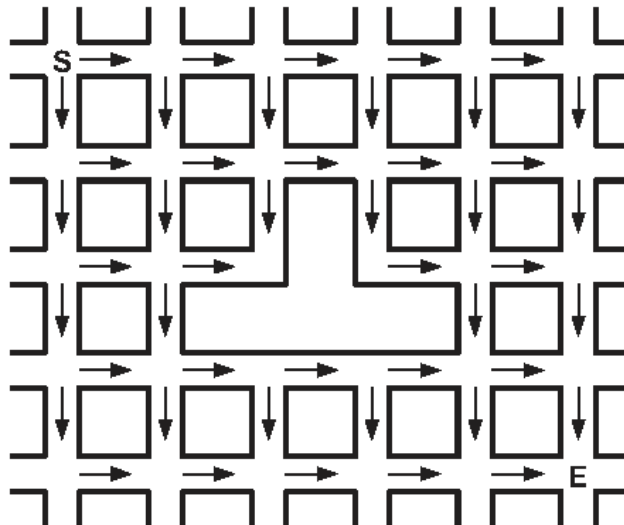


Figure B.



57B9A00C-55CE-45C5-8BF7-A73C94923445

SPRING2014_CSCI570_EXAM2

#40 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

DEE74E60-FEC1-4DEF-887B-52CC5E0792D2

SPRING2014_CSCI570_EXAM2

#40 9 of 10



Additional Space



EC4649FF-8594-41B9-B13E-9F86728618CC

SPRING2014_CSCI570_EXAM2

#40 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



76F8573B-41F8-4E03-9211-2BCCE5D59494

SPRING2014_CSCI570_EXAM2

#41 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



FE5E5436-1D9E-44CE-B30D-4CBF40A96F20

SPRING2014_CSCI570_EXAM2

#41 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

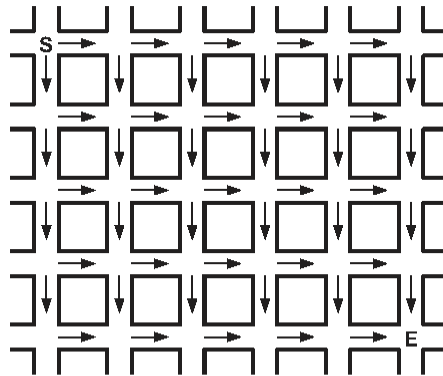


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

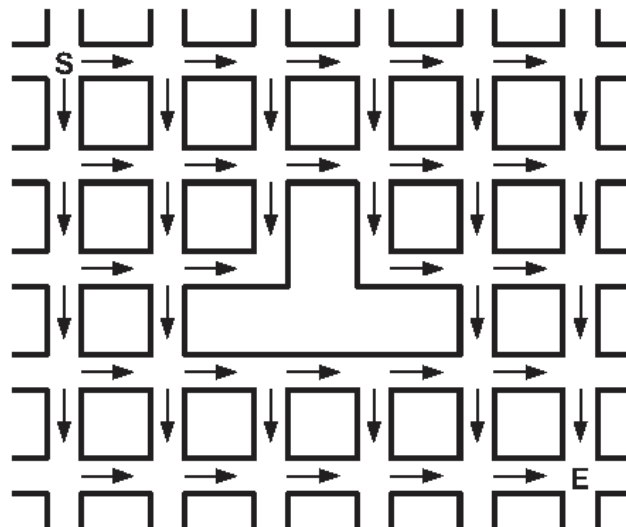


Figure B.



DD2D1206-373D-4852-8F3E-98C1A6BC0936

SPRING2014_CSCI570_EXAM2

#41 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

38797E0B-A56C-4E82-B91D-506D76F13427

SPRING2014_CSCI570_EXAM2

#41

9 of 10



Additional Space



AF1C0B3E-FDFF-4DBA-AC87-98F068EDF774

SPRING2014_CSCI570_EXAM2

#41 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



7B4CC8AD-5652-4E1F-BAD-C84B1D0256A6

SPRING2014_CSCI570_EXAM2

#42 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



90DF8DD2-B9E9-49EC-AEDE-A0AB0B6D7226

SPRING2014_CSCI570_EXAM2

#42 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

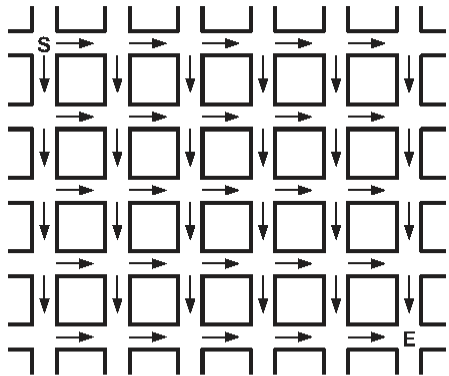


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

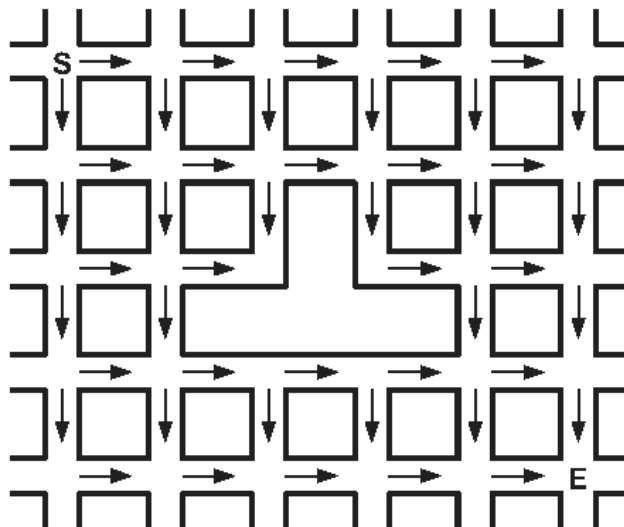


Figure B.



FE7B9913-B3CD-47B2-B92F-032CFB1EC9A3

SPRING2014_CSCI570_EXAM2

#42 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

EB575162-AB87-4BE9-8C6F-ECDBACD42E2C

SPRING2014_CSCI570_EXAM2

#42 9 of 10



Additional Space



19473958-B0DC-44C0-996D-BBB3C3A24211

SPRING2014_CSCI570_EXAM2

#42 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



F4C8E166-DEDD-432E-BA0C-5D9038256AF4

SPRING2014_CSCI570_EXAM2

#43 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



E4E284DC-4D7A-4CF2-A2F4-2D4F42F7565B

SPRING2014_CSCI570_EXAM2

#43 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

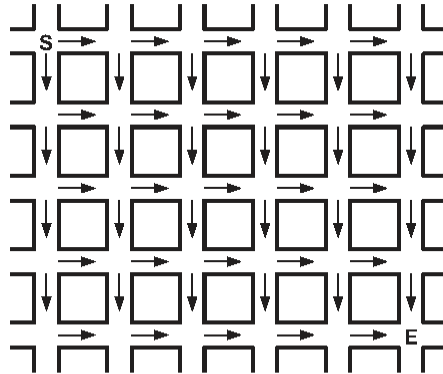


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

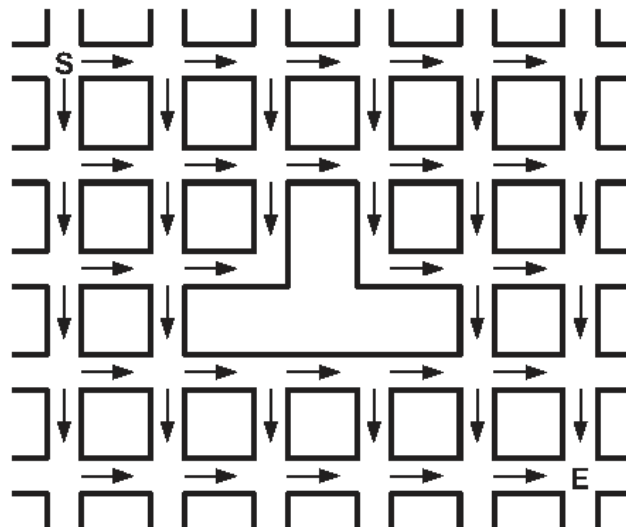


Figure B.



C647144B-0D32-4D47-89B4-C3D7BFB202DF

SPRING2014_CSCI570_EXAM2

#43 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

A32A3E3D-3390-4AD1-9D91-F5184D004BED

SPRING2014_CSCI570_EXAM2

#43 9 of 10



Additional Space



3C5B0607-A9A4-41A0-A6D9-A58D09F9D878

SPRING2014_CSCI570_EXAM2

#43 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



8898C35B-664B-4615-B6F2-49E11F3BE376

SPRING2014_CSCI570_EXAM2

#44 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



5) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\cdot, \cdot)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\cdot, \cdot)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

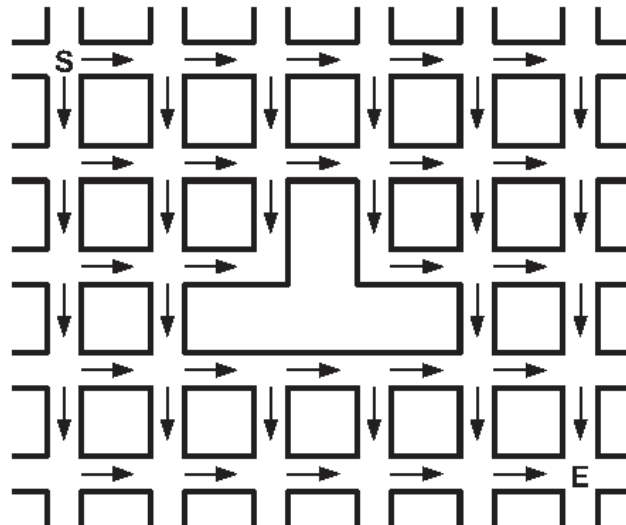


Figure B.



16FC71E9-1922-4B00-8CBF-5ABA56F4E879

SPRING2014_CSCI570_EXAM2

#44 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

2D8AC85F-5B47-46EA-AEC3-B389C5E6104C

SPRING2014_CSCI570_EXAM2

#44

9 of 10



Additional Space



517A1E55-914A-4825-A406-777656CAC23F

SPRING2014_CSCI570_EXAM2

#44 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



2AB7DF9E-DF37-4200-855C-6043F0C52263

SPRING2014_CSCI570_EXAM2

#45

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



4F513499-9004-427D-9DBB-70D58321777D

SPRING2014_CSCI570_EXAM2

#45

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

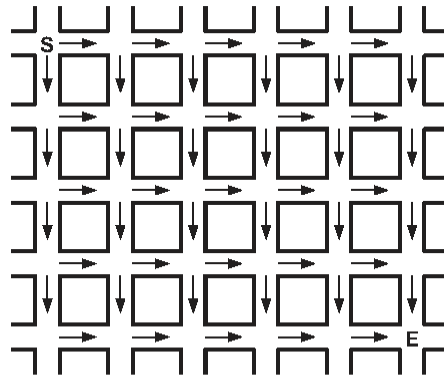


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

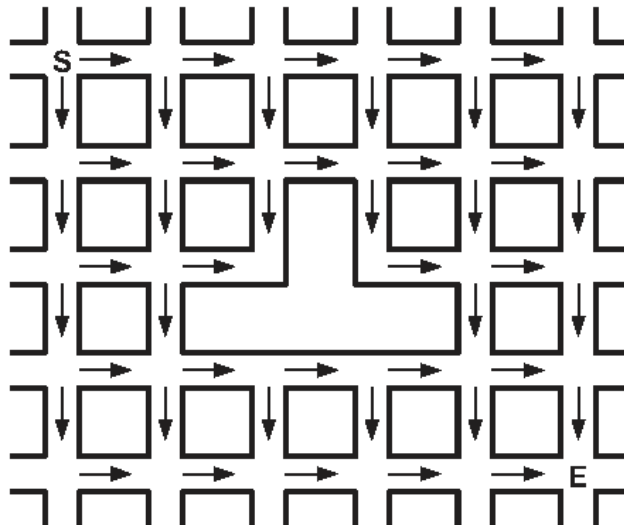


Figure B.



C4401E38-3286-4DF0-A29C-27C2CF3E2ECD

SPRING2014_CSCI570_EXAM2

#45 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

B0B20770-666F-4B5E-B9E4-325BFD4A32F3

SPRING2014_CSCI570_EXAM2

#45 9 of 10



Additional Space



1FB3BD6A-C8BF-491A-9A87-E417FA90154B

SPRING2014_CSCI570_EXAM2

#45 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



6AEDD59C-A5DE-4D7A-ACC0-84049DF88B30

SPRING2014_CSCI570_EXAM2

#46

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



2B7F77AD-A809-4DC0-9BC3-48C2C77CE2F1

SPRING2014_CSCI570_EXAM2

#46

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

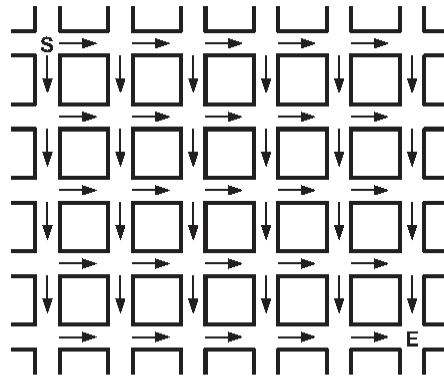


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

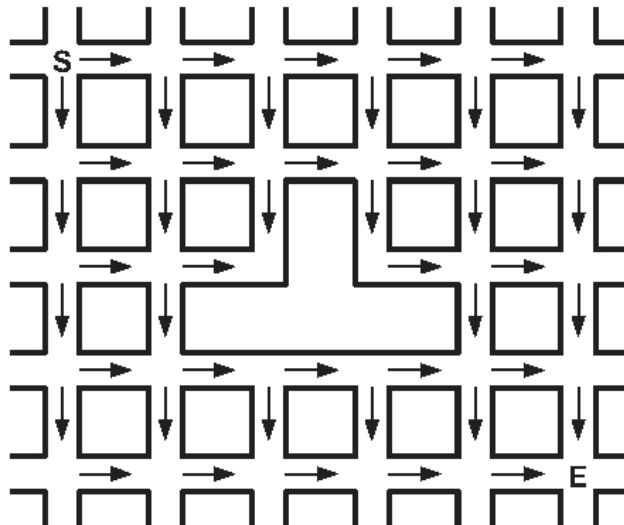


Figure B.



3A1FCA7A-D0C2-462A-B703-C8552F1F794E

SPRING2014_CSCI570_EXAM2

#46 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9AFB679B-43F8-4EE6-B3A5-AB128F8E608E

SPRING2014_CSCI570_EXAM2

#46 9 of 10



Additional Space



48F0C4CA-FA29-4FF8-B9BF-28989D88F22C

SPRING2014_CSCI570_EXAM2

#46 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



958EFC5F-5F70-4B4E-B46D-E5A6B2233216

SPRING2014_CSCI570_EXAM2

#47 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



836B7FC2-81FF-4464-9BBA-C033231EDAD6

SPRING2014_CSCI570_EXAM2

#47 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

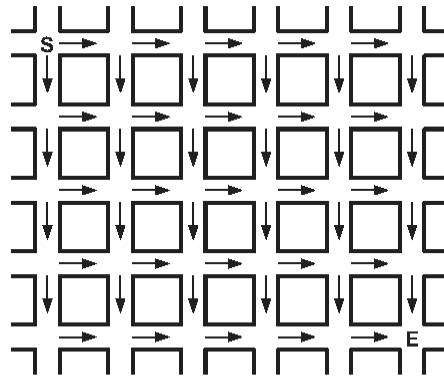


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



1CD7BE0C-8C18-4DBE-AE84-5946A8A890A2

SPRING2014_CSCI570_EXAM2

#47 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

00F55B3E-09C1-4A93-86D5-DBC7FAE526DE

SPRING2014_CSCI570_EXAM2

#47 9 of 10



Additional Space



8FB68F6D-D372-4019-824F-B0229FA0787A

SPRING2014_CSCI570_EXAM2

#47 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



542B6CB9-DE95-49DE-960C-22A52F2FD2DC

SPRING2014_CSCI570_EXAM2

#48

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



22433CE0-65C0-496D-97D6-D62E7CB989AA

SPRING2014_CSCI570_EXAM2

#48 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

F3778B54-56D5-4CAB-8E93-4FB8557EF3A1

SPRING2014_CSCI570_EXAM2

#48

9 of 10



Additional Space



841D3792-9FED-45A9-BD3B-DA92594B00B5

SPRING2014_CSCI570_EXAM2

#48 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



83C41FA3-7E28-4436-A8B2-B9D61FCD2C0E

SPRING2014_CSCI570_EXAM2

#49

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

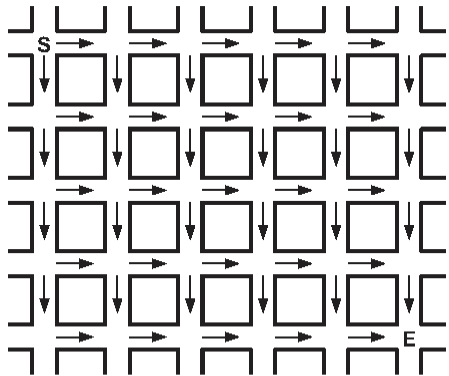


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



E62E6C5B-C6B0-45AA-8DD7-1EC7A34D37E4

SPRING2014_CSCI570_EXAM2

#49 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

1A8DCEB6-75F1-454F-B9FB-366A89CFFF67

SPRING2014_CSCI570_EXAM2

#49 9 of 10



Additional Space



5B5EE595-6E4F-46C8-8910-7321A954E53A

SPRING2014_CSCI570_EXAM2

#49 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



34018719-4190-4A33-9AA0-61A91337272A

SPRING2014_CSCI570_EXAM2

#50 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

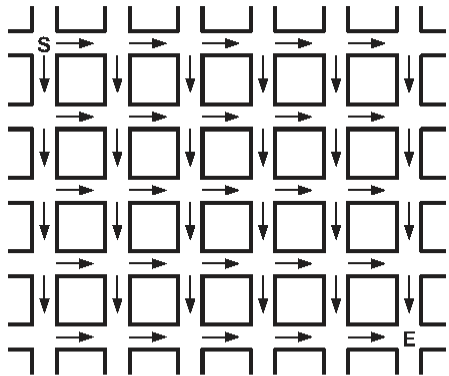


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

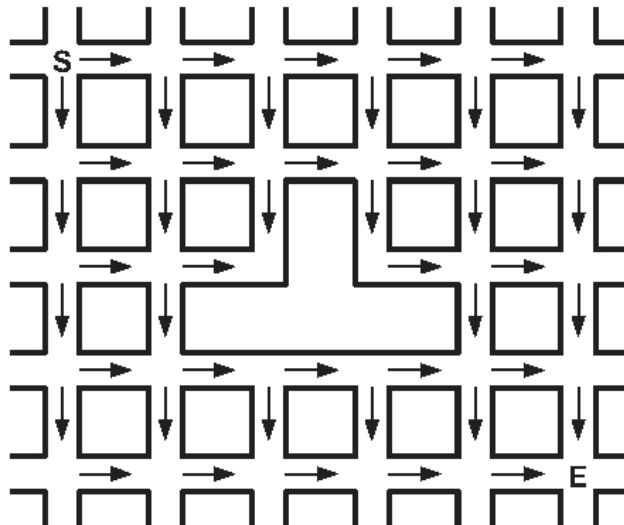


Figure B.



D6CBF984-423C-453B-A93F-C9FDE63D5E27

SPRING2014_CSCI570_EXAM2

#50 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

E1CACA32-BDD9-4451-B680-83CD740849C5

SPRING2014_CSCI570_EXAM2

#50

9 of 10



Additional Space



48308734-E473-49C3-BFCE-0E4C96E11415

SPRING2014_CSCI570_EXAM2

#50 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



1D9B8389-DA4D-4ACC-B090-2F24172B28CA

SPRING2014_CSCI570_EXAM2

#51 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

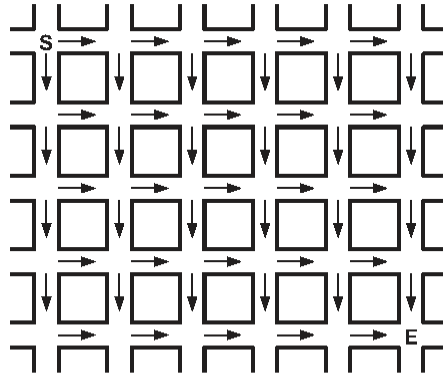


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

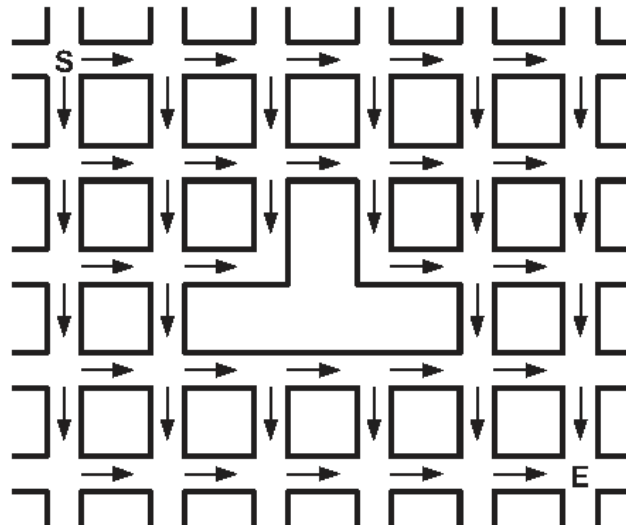


Figure B.



644A2643-FDFA-4B75-A503-07E02EE83CA6

SPRING2014_CSCI570_EXAM2

#51 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

3EE9EF25-0511-4E83-A705-0924CF441C8A

SPRING2014_CSCI570_EXAM2

#51

9 of 10



Additional Space



DA8FB917-59D8-4BC2-9F51-BBF6C6878422

SPRING2014_CSCI570_EXAM2

#51 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



C71C3577-E2C2-499D-A7F6-DBA27DA6F43E

SPRING2014_CSCI570_EXAM2

#52 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



C4F48E3D-ABE2-492E-9836-A8FADAF620E

SPRING2014_CSCI570_EXAM2

#52

6 of 10

5) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

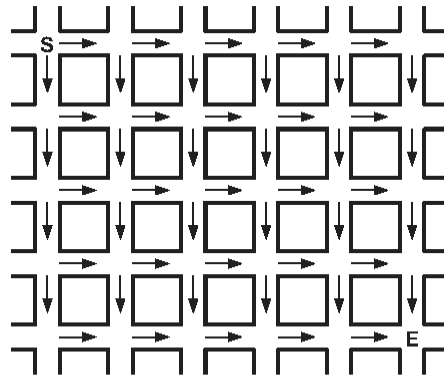


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



F8FC5AC7-2DA3-48A9-9372-28CAC7F06920

SPRING2014_CSCI570_EXAM2

#52 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

5C194E17-6845-4205-882F-E35C7813F36B

SPRING2014_CSCI570_EXAM2

#52 9 of 10



Additional Space



A47FFFAF-93B3-49A0-BCD3-0D27E4F0ED5E

SPRING2014_CSCI570_EXAM2

#52 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



79B017A1-B2D5-4DAD-B7CA-7CDEB02BA399

SPRING2014_CSCI570_EXAM2

#53 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

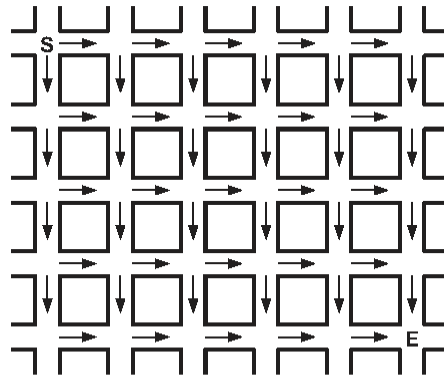


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



055F957E-ADE2-48A6-8EED-0CC6C3DB3ADD

SPRING2014_CSCI570_EXAM2

#53 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

95A80C7F-1598-4352-93CE-6ABD37E193C5

SPRING2014_CSCI570_EXAM2

#53

9 of 10



Additional Space



8AA36CB3-1F5F-43D2-A368-13A737F5A702

SPRING2014_CSCI570_EXAM2

#53 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



016B2413-011D-46AF-9D48-CF5FCD84B6E3

SPRING2014_CSCI570_EXAM2

#54 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

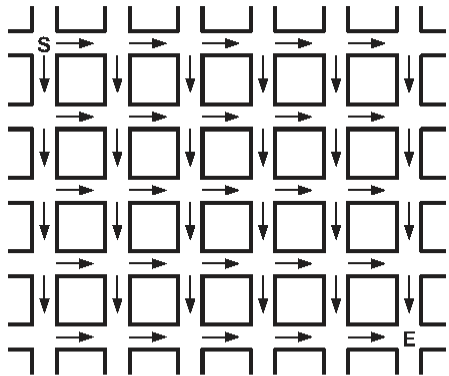


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



486ED0B2-5EE5-4FE5-AB63-4D86FD3E51F2

SPRING2014_CSCI570_EXAM2

#54 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

21E41817-FBFC-4308-A08E-8625D519E268

SPRING2014_CSCI570_EXAM2

#54

9 of 10



Additional Space



673811BC-2F04-4F11-9DC7-160D8D422509

SPRING2014_CSCI570_EXAM2

#54 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



4E97B4E5-90B0-4CC0-AFE9-12E479E8CBDD

SPRING2014_CSCI570_EXAM2

#55

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

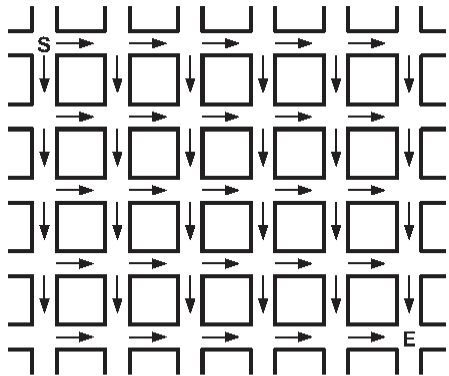


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

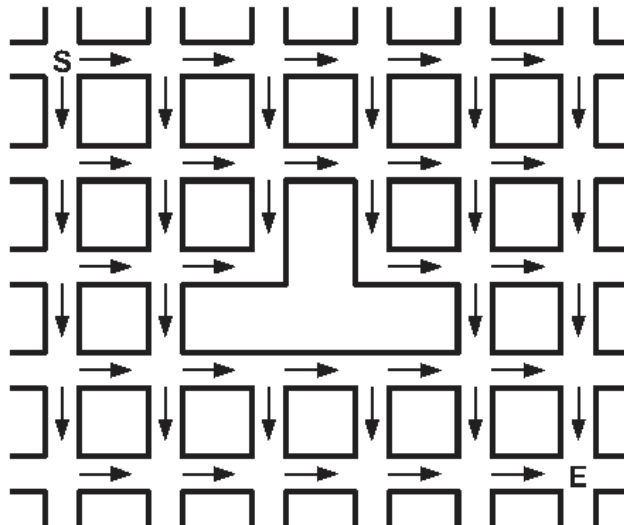


Figure B.



67440589-E1E0-4914-BAA5-B512BD69C3E6

SPRING2014_CSCI570_EXAM2

#55 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

A6B7DC13-B198-40A1-9554-033663900331

SPRING2014_CSCI570_EXAM2

#55

9 of 10



Additional Space



929E1F28-A119-4B15-9CB9-A9238F423E41

SPRING2014_CSCI570_EXAM2

#55 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



012B55B2-261A-4126-80CD-F0A3F92B3870

SPRING2014_CSCI570_EXAM2

#56 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



989CAD0C-B09F-48EC-BC3B-029921799D63

SPRING2014_CSCI570_EXAM2

#56 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

699470BB-14CA-440E-84F5-72E06BC3DCE1

SPRING2014_CSCI570_EXAM2

#56 9 of 10



Additional Space



C4425DE5-1981-4A66-8CD8-76AFCA39CF77

SPRING2014_CSCI570_EXAM2

#56 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



56F49050-96BF-485A-AC26-CC3F6C21A22E

SPRING2014_CSCI570_EXAM2

#57 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

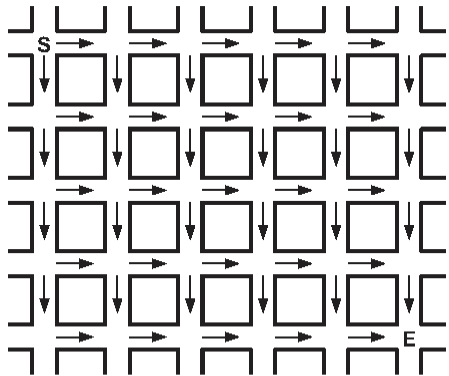


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



69447D7B-BD00-426A-8EA7-75F9ED32B337

SPRING2014_CSCI570_EXAM2

#57 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

4F130E78-D6D6-4243-B544-C75F559339C7

SPRING2014_CSCI570_EXAM2

#57

9 of 10



Additional Space



13CC13E8-E894-4BB7-BA39-F23AEC0BACC4

SPRING2014_CSCI570_EXAM2

#57 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



9B17F6FD-2B5A-4819-BA4A-7B143FBB7DBB

SPRING2014_CSCI570_EXAM2

#58 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

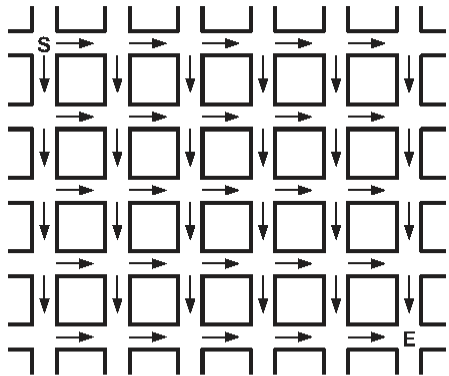


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



FE166D8F-77AA-4C62-9FB5-7CD821729C54

SPRING2014_CSCI570_EXAM2

#58 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

02A1BBC0-55C7-40CF-906C-7B4D0B0800A5

SPRING2014_CSCI570_EXAM2

#58

9 of 10



Additional Space



310E2437-E544-4CC1-B353-F5A58582BEB8

SPRING2014_CSCI570_EXAM2

#58 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



0C26941F-9D74-4AC2-B9D3-149FB31B8397

SPRING2014_CSCI570_EXAM2

#59

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

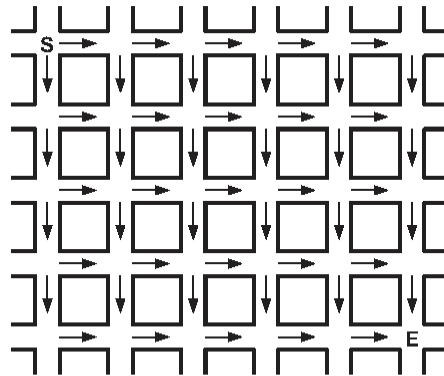


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

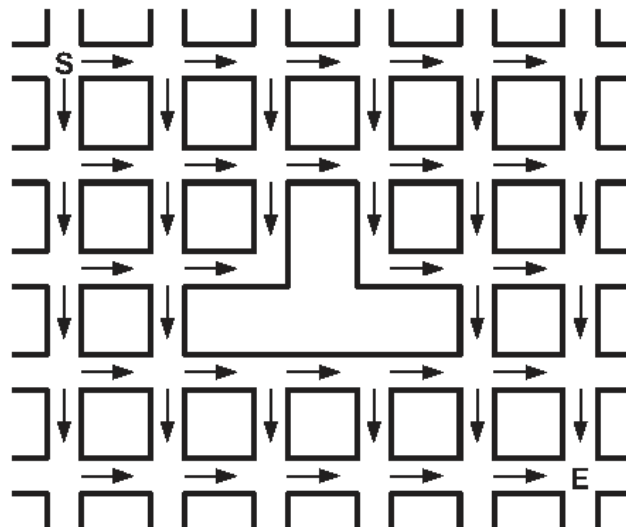


Figure B.



7D3B9ABA-BB3E-4506-890C-D1352D65AC4F

SPRING2014_CSCI570_EXAM2

#59 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

00E99A50-4316-4ECE-A3C2-77788249D40E

SPRING2014_CSCI570_EXAM2

#59 9 of 10



Additional Space



0E6B3BBC-9F16-466D-BBC3-7BE374848F08

SPRING2014_CSCI570_EXAM2

#59

10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



9A495868-F473-4937-B7C8-EEF7068E2DE6

SPRING2014_CSCI570_EXAM2

#60 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



02CAF5FC-3EFE-42A8-AA56-EE4D00B68C36

SPRING2014_CSCI570_EXAM2

#60 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

43C0F558-3C17-484F-866B-8D1C47B881FC

SPRING2014_CSCI570_EXAM2

#60

9 of 10



Additional Space



3F2CC8F6-C3E5-4F2B-A58E-98F07728933A

SPRING2014_CSCI570_EXAM2

#60 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



B37823E6-5BA8-4FAE-8900-3A4BC8B6FF51

SPRING2014_CSCI570_EXAM2

#61 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



6CAAA519-DA38-4C59-84B8-5BD8B45EA8D7

SPRING2014_CSCI570_EXAM2

#61 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



61AC7343-D26D-45B9-A7CF-32284ABC6428

SPRING2014_CSCI570_EXAM2

#61

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

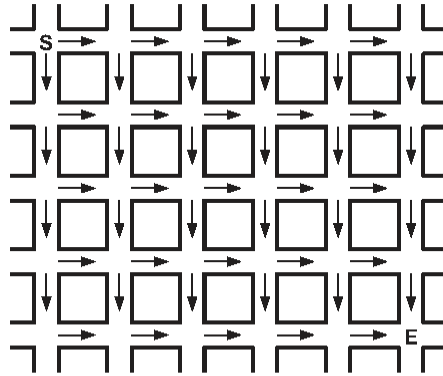


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

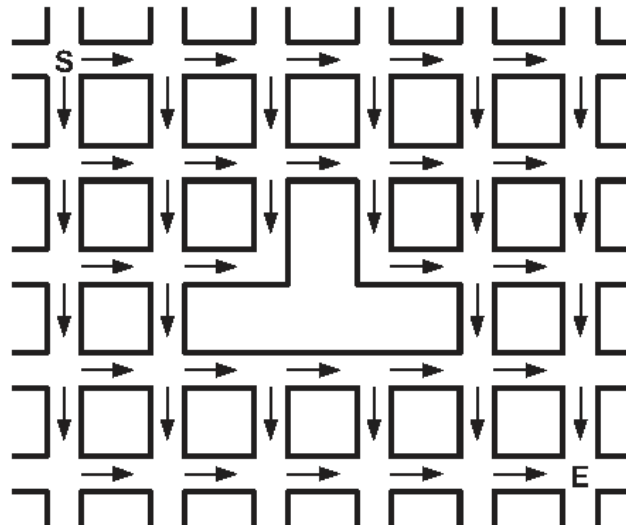


Figure B.



0607A576-256E-423C-97C6-BA9A49E828E1

SPRING2014_CSCI570_EXAM2

#61 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

0C4212D8-DF4E-43D0-87BC-7BE70891F5D7

SPRING2014_CSCI570_EXAM2

#61 9 of 10



Additional Space



0E61E1EB-FC0C-448C-931D-EB2034CA8C55

SPRING2014_CSCI570_EXAM2

#61 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



E14F8E51-EB42-4CA1-A29F-E838DF77A92C

SPRING2014_CSCI570_EXAM2

#62 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



D01A3B33-D88C-417D-970A-CCBA9FBA7DA8

SPRING2014_CSCI570_EXAM2

#62 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



D51F1B65-69EC-4DB4-A73F-5C92DB296E0E

SPRING2014_CSCI570_EXAM2

#62 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

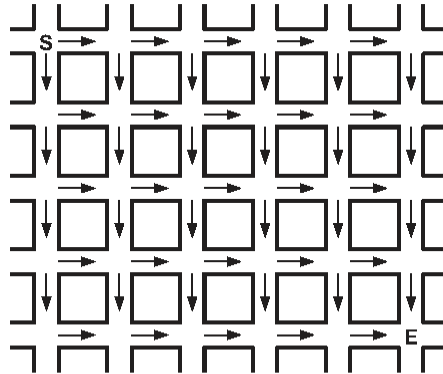


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



1D2A9E72-A1C0-4C5E-8CA4-F2A922BBA816

SPRING2014_CSCI570_EXAM2

#62 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

E84B3086-9EEE-4DBA-80EB-9BDF53A520BB

SPRING2014_CSCI570_EXAM2

#62 9 of 10



Additional Space



0364DA3E-94EF-4457-9009-80E8B44A2A58

SPRING2014_CSCI570_EXAM2

#62 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



7269995C-F706-4221-9F29-FA2B1D664574

SPRING2014_CSCI570_EXAM2

#63 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



17D67D70-47FE-4776-9A01-AC0513A1AE53

SPRING2014_CSCI570_EXAM2

#63 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

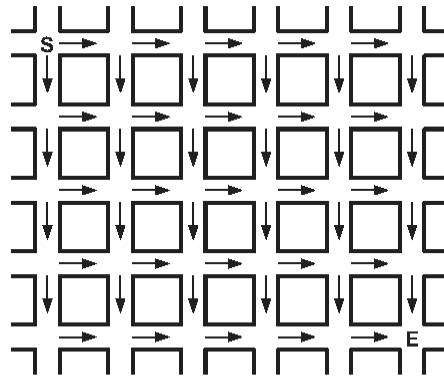


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



10EAD7E9-E1D3-47DD-BE87-4B9A6272AE7F

SPRING2014_CSCI570_EXAM2

#63 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

7BE43B0B-EF33-42B0-8746-91247BBF1663

SPRING2014_CSCI570_EXAM2

#63 9 of 10



Additional Space



43B8BD86-B1E9-43B9-80D4-DF05A3C41812

SPRING2014_CSCI570_EXAM2

#63 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



B182AC22-C579-41C2-A12D-C1F68FAEEC44

SPRING2014_CSCI570_EXAM2

#64 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

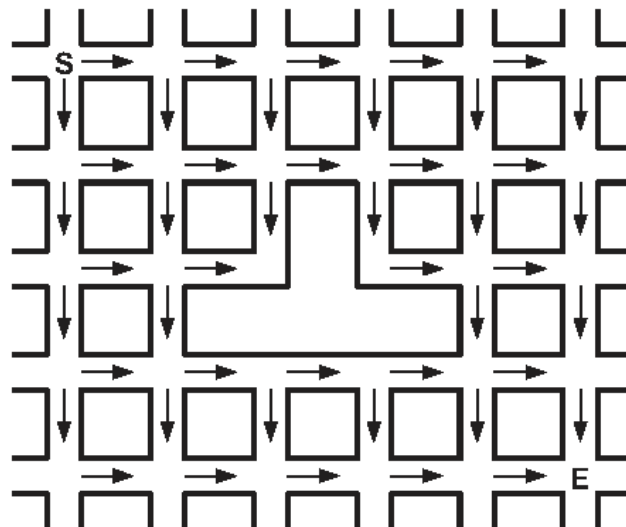


Figure B.



20F817ED-BBA8-4F0D-9F11-2B3DDDF300F1

SPRING2014_CSCI570_EXAM2

#64 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

32323C92-6435-4B93-9195-20AC1DADC78C

SPRING2014_CSCI570_EXAM2

#64

9 of 10



Additional Space



58ACBC6D-E6F9-4A10-AF03-DB1F50EC7B0E

SPRING2014_CSCI570_EXAM2

#64 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



BA8ED1F9-0200-4FC3-8ECB-D10F1102A594

SPRING2014_CSCI570_EXAM2

#65 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



ABBAEA6F-171B-46BD-B7E3-251FFF8B5244

SPRING2014_CSCI570_EXAM2

#65

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

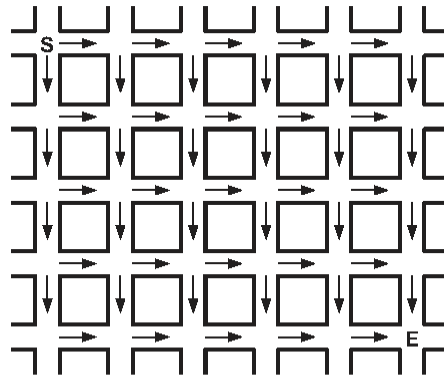


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

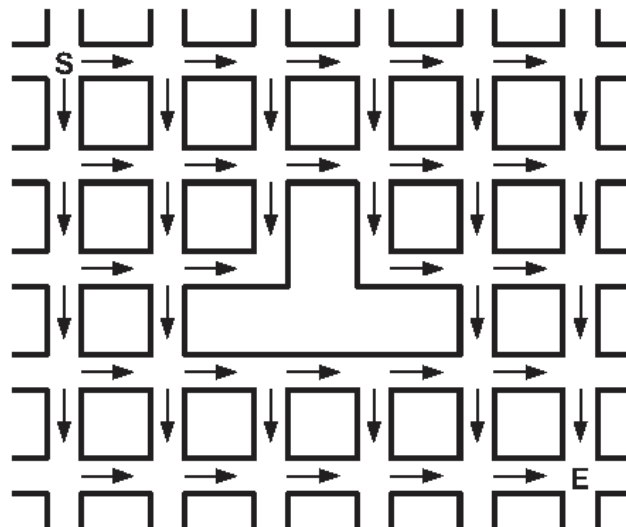


Figure B.



59B332EB-F7CB-48E9-B651-35163C348CD0

SPRING2014_CSCI570_EXAM2

#65 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

0E228720-DD61-49A8-89E5-82B79B58343E

SPRING2014_CSCI570_EXAM2

#65 9 of 10



Additional Space



A7692E2C-F2A2-494E-9AF8-30D1149432C1

SPRING2014_CSCI570_EXAM2

#65 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



DB5940F0-77B2-403A-A7C7-DDD2B95DEE15

SPRING2014_CSCI570_EXAM2

#66 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



323D7022-641B-44A2-87ED-017FCBFDAFFF

SPRING2014_CSCI570_EXAM2

#66

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

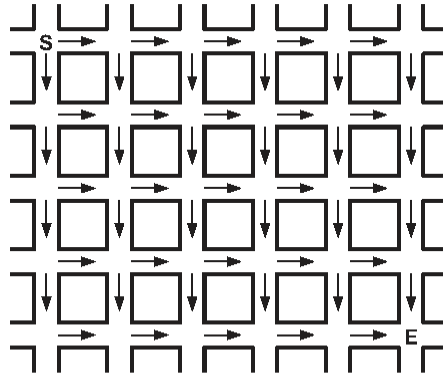


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



53668018-AD45-4099-97F2-CC38F3C93E04

SPRING2014_CSCI570_EXAM2

#66 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

B93FE20E-D745-42D6-80ED-E2E8011A327E

SPRING2014_CSCI570_EXAM2

#66 9 of 10



Additional Space



A931FD02-2D05-40A5-A8D5-AAFD56BE1B7F

SPRING2014_CSCI570_EXAM2

#66 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



3FFFADE3-4A56-4FD4-9F80-3BADE2F647DF

SPRING2014_CSCI570_EXAM2

#67 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



6DE41999-F188-4742-84DE-0773B3D551EC

SPRING2014_CSCI570_EXAM2

#67

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

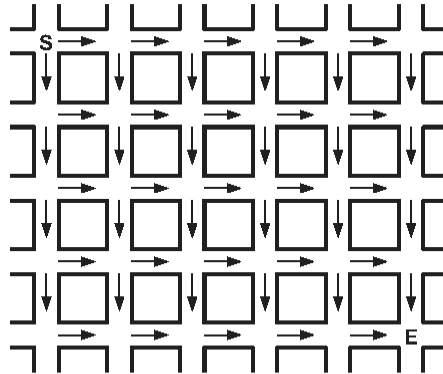


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

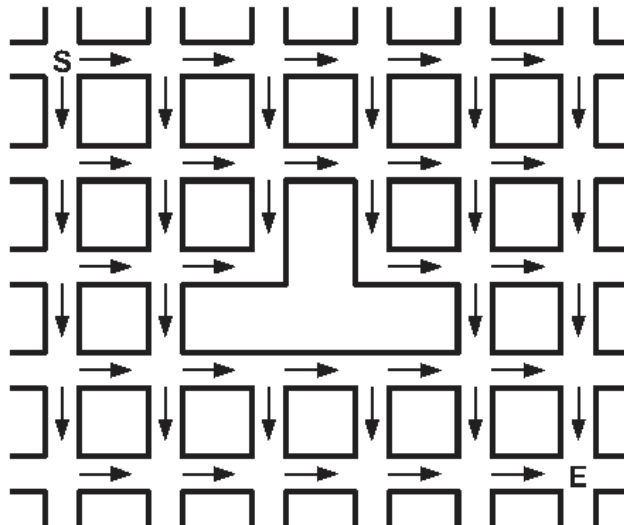


Figure B.



E11A0E1C-0444-4396-B09C-4044C11AEA6A

SPRING2014_CSCI570_EXAM2

#67 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

ADB37D07-7A10-4A74-ABE8-56875DA1069F

SPRING2014_CSCI570_EXAM2

#67 9 of 10



Additional Space



5DBECB7F-16D3-45ED-AB40-3E162001AE4F

SPRING2014_CSCI570_EXAM2

#67 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



2B4E65DB-8312-476A-BC7C-E3E616F5B1D3

SPRING2014_CSCI570_EXAM2

#68

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

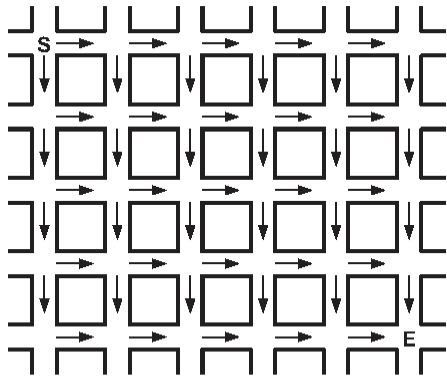


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



46201C36-D824-49DE-B24D-72CC78F29AF3

SPRING2014_CSCI570_EXAM2

#68 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

C70CDFC8-DB73-43DF-A136-EB3DEC3A246A

SPRING2014_CSCI570_EXAM2

#68

9 of 10



Additional Space



5757B60F-4C2C-4EC2-80E6-9098FCF1918E

SPRING2014_CSCI570_EXAM2

#68 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



BA196021-E1CC-478B-A023-54CBE114A0E1

SPRING2014_CSCI570_EXAM2

#69

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



2BCEF69E-43ED-43CF-B335-C2794B05944B

SPRING2014_CSCI570_EXAM2

#69

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

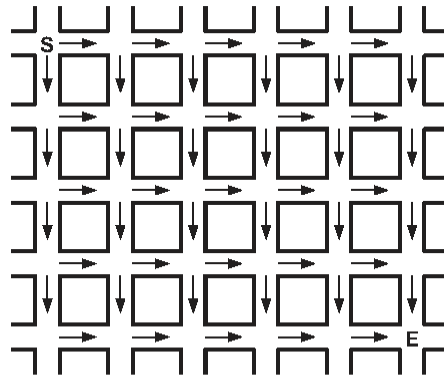


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

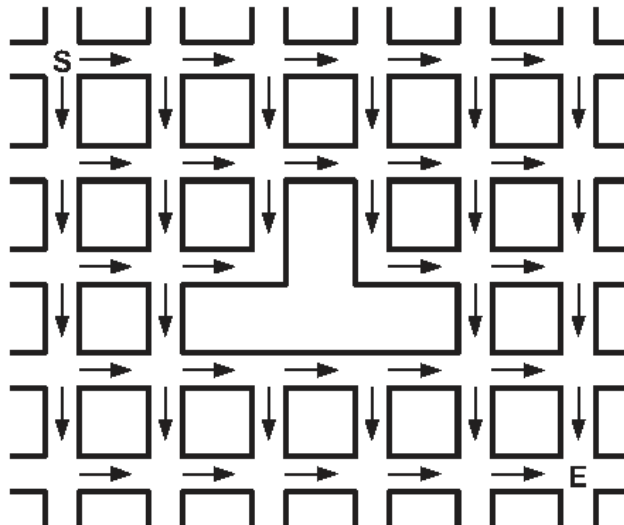


Figure B.



0E2B1352-D3E8-44E6-A979-7B9B56369EA3

SPRING2014_CSCI570_EXAM2

#69 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

51B4B312-ACB2-4C9F-81CE-C089F992AB06

SPRING2014_CSCI570_EXAM2

#69

9 of 10



Additional Space



42F0C0AB-FB1E-4754-A465-CF1328EC3EFA

SPRING2014_CSCI570_EXAM2

#69 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



2A84E6E4-2FE9-42AB-84CA-83A7AA040730

SPRING2014_CSCI570_EXAM2

#70

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

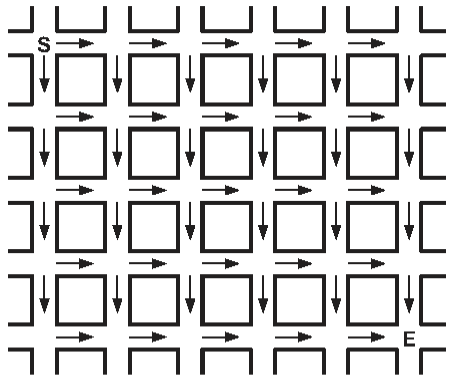


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



8A9C7A5C-7CC0-4C61-A056-C53301E80658

SPRING2014_CSCI570_EXAM2

#70 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

1F649F0A-6642-4B86-9A8A-B412649D0654

SPRING2014_CSCI570_EXAM2

#70 9 of 10



Additional Space



A6ED2A5E-4D66-4BA6-97A3-89422C52AAEA

SPRING2014_CSCI570_EXAM2

#70 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



74769C9C-D84E-4677-A4E6-8530BE1C4683

SPRING2014_CSCI570_EXAM2

#71 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

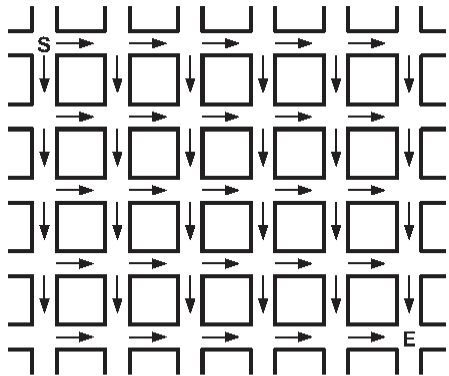


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



D3B0F5E0-EAD3-4ED9-8088-869F21FDFD97

SPRING2014_CSCI570_EXAM2

#71 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

C5311FBA-7DCB-417E-802C-54F77D3AD926

SPRING2014_CSCI570_EXAM2

#71

9 of 10



Additional Space



7EEB07F9-5F58-4DCD-85B6-C7FF5BC60307

SPRING2014_CSCI570_EXAM2

#71 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



1E09D74A-F033-4E96-804A-BC61BA29B453

SPRING2014_CSCI570_EXAM2

#72 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

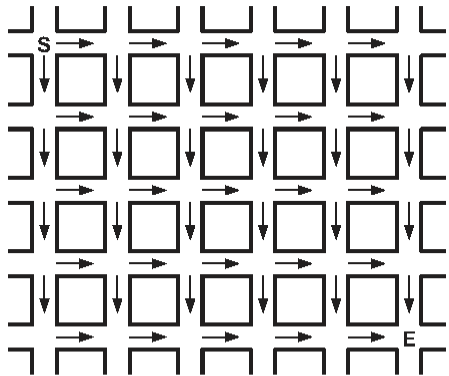


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



544A504E-FB11-4069-A597-1BEC7E56860E

SPRING2014_CSCI570_EXAM2

#72 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

CF1573C6-EFF6-47DB-B7FB-1AF26999BB96

SPRING2014_CSCI570_EXAM2

#72

9 of 10



Additional Space



3D7DAC81-919E-4074-853F-1DC0B0D9469D

SPRING2014_CSCI570_EXAM2

#72 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



E142EE8B-E87A-4021-85FD-9DDA7FD409B2

SPRING2014_CSCI570_EXAM2

#73 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



742AAFC1-740D-472E-81E3-B0E73705D2BC

SPRING2014_CSCI570_EXAM2

#73

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

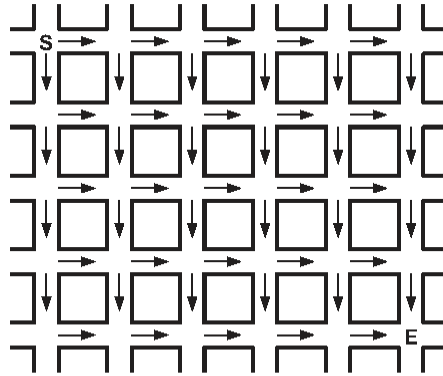


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

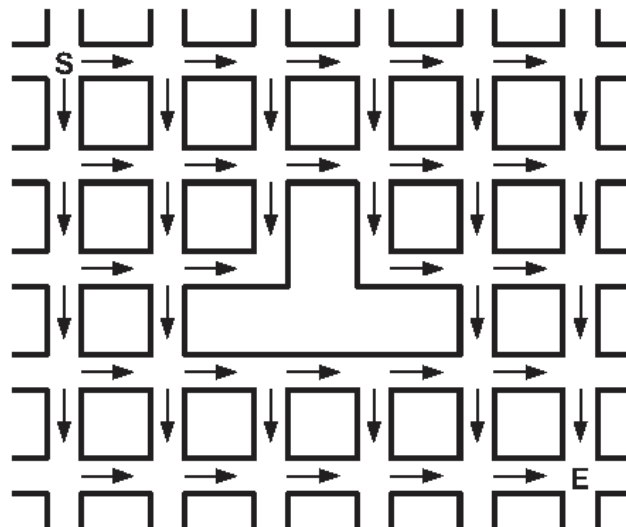


Figure B.



B6348515-A66E-44E8-91E9-C424623874D6

SPRING2014_CSCI570_EXAM2

#73 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

32B38629-0DEA-456C-9F2E-2A5CE1B46719

SPRING2014_CSCI570_EXAM2

#73

9 of 10



Additional Space



E4BA5FEC-5490-4941-86C4-34E5AE238876

SPRING2014_CSCI570_EXAM2

#73 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



FD36F378-85FB-40AF-A38F-7D4E287E191C

SPRING2014_CSCI570_EXAM2

#74 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

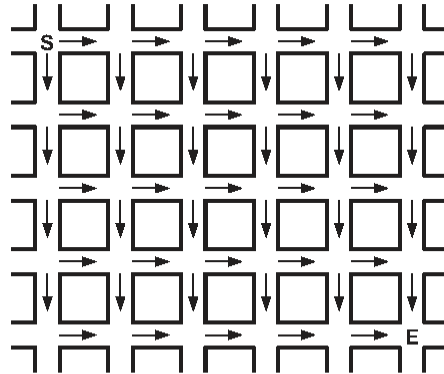


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

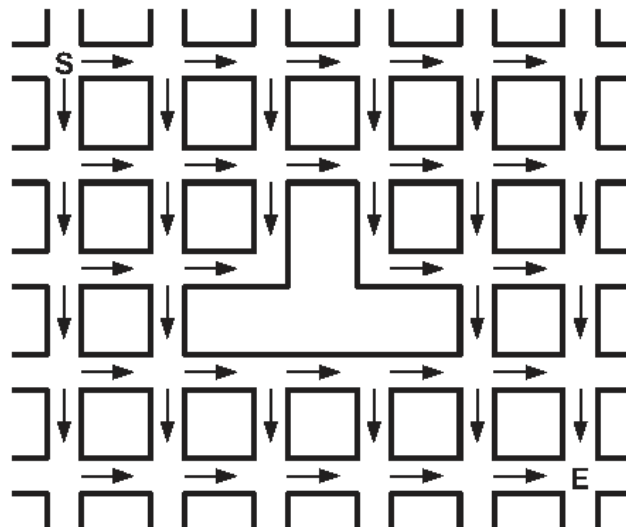


Figure B.



5351E9BF-4222-43F2-B84A-513977C8080D

SPRING2014_CSCI570_EXAM2

#74 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

E8EA7151-7E75-49D7-BA14-069739A8CF06

SPRING2014_CSCI570_EXAM2

#74

9 of 10



Additional Space



BEDF659A-9741-4595-8906-390A4F495D66

SPRING2014_CSCI570_EXAM2

#74 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



3A5E7DED-785A-4873-BA57-9966E6C51E1C

SPRING2014_CSCI570_EXAM2

#75 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



4CD1BA44-7452-4AB6-B29F-7EDA93862CB2

SPRING2014_CSCI570_EXAM2

#75

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

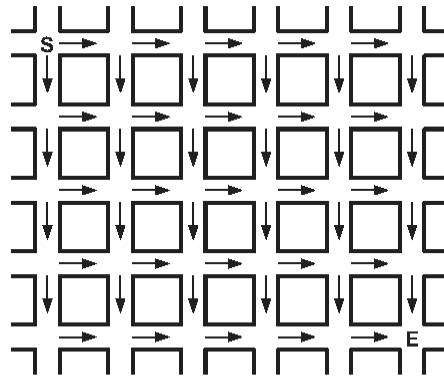


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



05CE477E-705C-451C-8E14-C8C0797A1F96

SPRING2014_CSCI570_EXAM2

#75 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

D3509CDE-8E2F-4DC8-AE16-05AC9ED0376C

SPRING2014_CSCI570_EXAM2

#75

9 of 10



Additional Space



657D490C-ED8D-4A97-9A1B-7A20ED59E095

SPRING2014_CSCI570_EXAM2

#75 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



807367C5-6249-4024-86C2-2B0A85A189CA

SPRING2014_CSCI570_EXAM2

#76 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



81BE854E-1123-4A7A-9A91-92F5D4ACEC8B

SPRING2014_CSCI570_EXAM2

#76 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

2B109E8C-CB35-4ADB-B86E-9B47933386F4

SPRING2014_CSCI570_EXAM2

#76 9 of 10



Additional Space



AA10BDD6-3762-4613-A371-85BE6B2B828C

SPRING2014_CSCI570_EXAM2

#76 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



217C0CF0-20E4-4E40-8FDF-A43DC70C9621

SPRING2014_CSCI570_EXAM2

#77 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



2CE66537-F092-4FE2-8885-8562E8761B8F

SPRING2014_CSCI570_EXAM2

#77

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

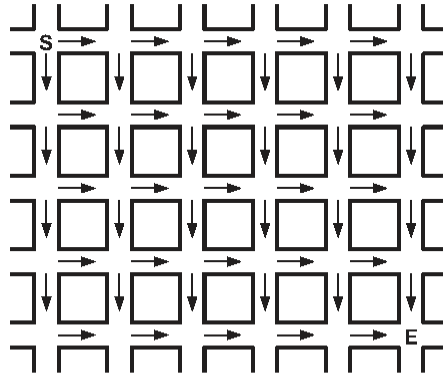


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

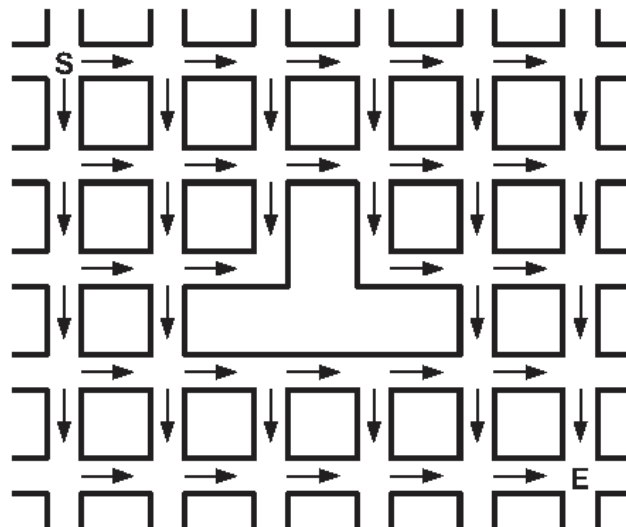


Figure B.



ED3113D3-0FB7-47C2-A2FE-EC72F55464E0

SPRING2014_CSCI570_EXAM2

#77 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

A8E6CC03-11C0-44A2-98D3-11CD3875A8C7

SPRING2014_CSCI570_EXAM2

#77

9 of 10



Additional Space



CB8D52B0-FDAF-4901-83F1-5B8343CE2470

SPRING2014_CSCI570_EXAM2

#77 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



DD5F3834-7E40-4642-87B1-1180B779138F

SPRING2014_CSCI570_EXAM2

#78 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



9FE2F4FC-17F0-468A-A305-EC65C586982D

SPRING2014_CSCI570_EXAM2

#78

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

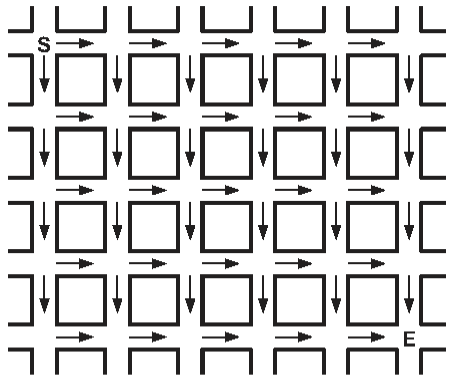


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

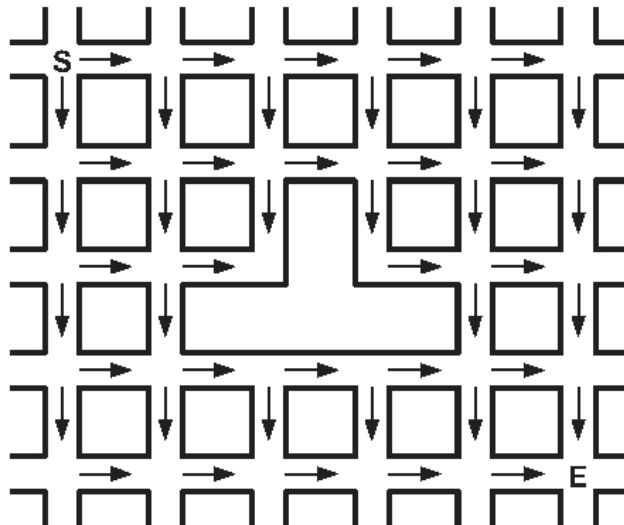


Figure B.



00896D7F-847E-47FA-B718-BB7A1F89827A

SPRING2014_CSCI570_EXAM2

#78 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

CD208984-C94A-4D99-82F3-698696E579D0

SPRING2014_CSCI570_EXAM2

#78

9 of 10



Additional Space



67421856-46EB-4BA3-83F9-94AD3FD3064E

SPRING2014_CSCI570_EXAM2

#78 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



80D14A66-5C60-47E4-A97E-48EDB2923D09

SPRING2014_CSCI570_EXAM2

#79

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

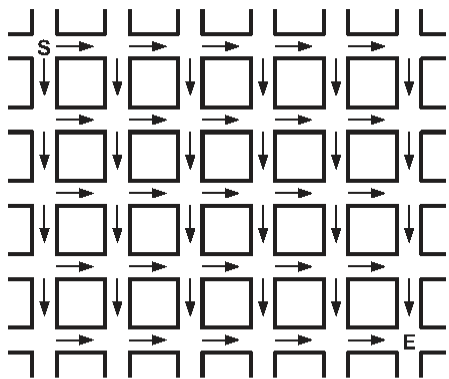


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

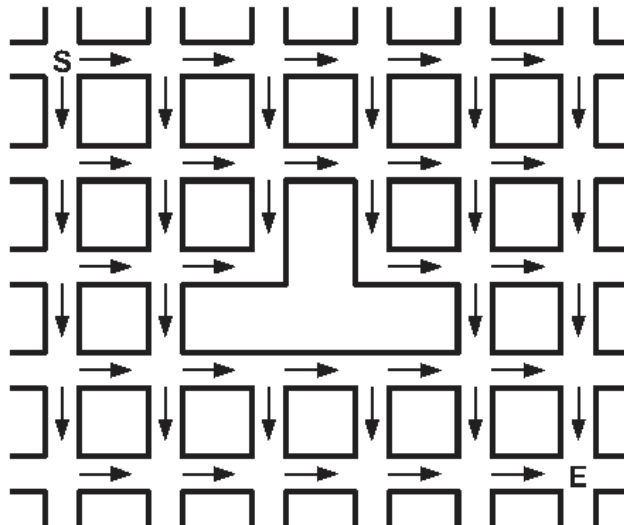


Figure B.



99CBD086-AE02-4A07-A53A-9CC2971FEAA1

SPRING2014_CSCI570_EXAM2

#79 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

BAC3AFD6-8FD7-452F-BC9F-44A9B6630B79

SPRING2014_CSCI570_EXAM2

#79

9 of 10



Additional Space



95850D06-736C-45C8-B61D-4DE247B35C2F

SPRING2014_CSCI570_EXAM2

#79 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1F9211BD-B0C7-4E6D-A181-EE7B3F8DE4E7

SPRING2014_CSCI570_EXAM2

#80 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



A31FF727-DA92-4F29-BA32-D112DC88A1A9

SPRING2014_CSCI570_EXAM2

#80 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



2410349F-4DC1-4D41-9F02-8B6119F2AAED

SPRING2014_CSCI570_EXAM2

#80

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

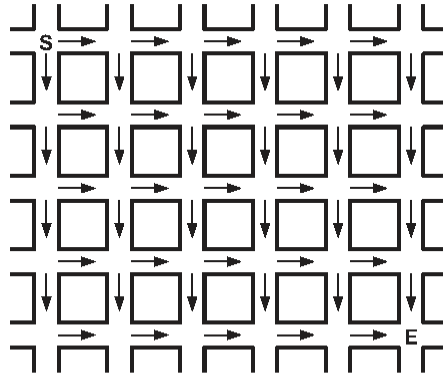


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

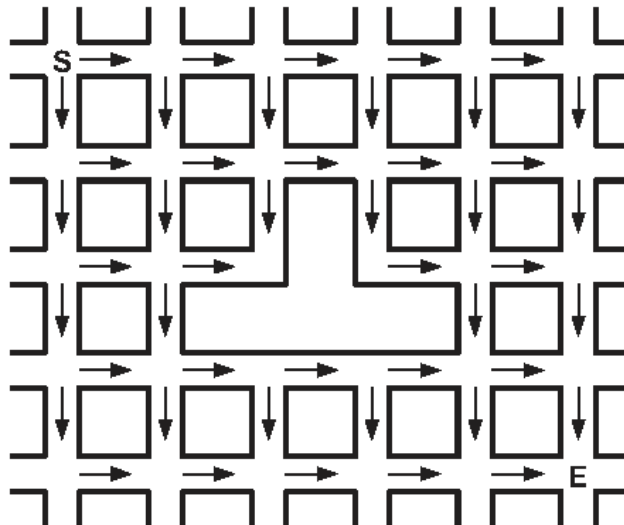


Figure B.



E50523E1-5BF4-4E81-80A1-DF516D79FE9B

SPRING2014_CSCI570_EXAM2

#80 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

96375397-F84C-46FD-8942-74E20BBA09AD

SPRING2014_CSCI570_EXAM2

#80 9 of 10



Additional Space



8472F00E-D454-45DB-B761-5F489DA37A4D

SPRING2014_CSCI570_EXAM2

#80 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



928E5143-E3D6-4CB6-BC79-FC2C8E08EDFA

SPRING2014_CSCI570_EXAM2

#81 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

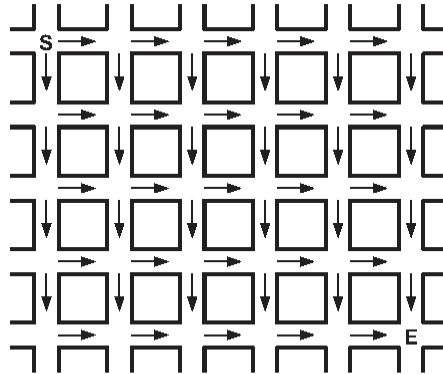


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



19E83A75-01D4-4B19-A5C0-C6AA3DCAAA63

SPRING2014_CSCI570_EXAM2

#81 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

80C5D924-762F-4CB3-92BB-1BAD064D3FF5

SPRING2014_CSCI570_EXAM2

#81 9 of 10



Additional Space



4BCE8444-BB76-40B3-80CC-D8957C0AEE8B

SPRING2014_CSCI570_EXAM2

#81 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



3730BDC1-8471-4796-B123-1C7CDE46C664

SPRING2014_CSCI570_EXAM2

#82 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



0859E1D1-B70B-4CC2-887C-90EF50801BB6

SPRING2014_CSCI570_EXAM2

#82

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

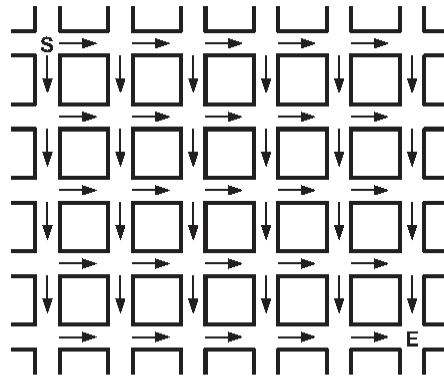


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



2A6AB5F1-7E7B-4890-9B57-2A3EBFF55120

SPRING2014_CSCI570_EXAM2

#82 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

EE2BC67D-900E-4AB1-BF3E-1E0AC65A856E

SPRING2014_CSCI570_EXAM2

#82

9 of 10



Additional Space



F26F2DB8-2FFE-487A-A478-F587C508CE83

SPRING2014_CSCI570_EXAM2

#82 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



BB7C9D29-7AFD-46BC-A014-92584DCB9BDD

SPRING2014_CSCI570_EXAM2

#83 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



A9179D96-A303-447E-BA6D-51E2F5B7D97E

SPRING2014_CSCI570_EXAM2

#83 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

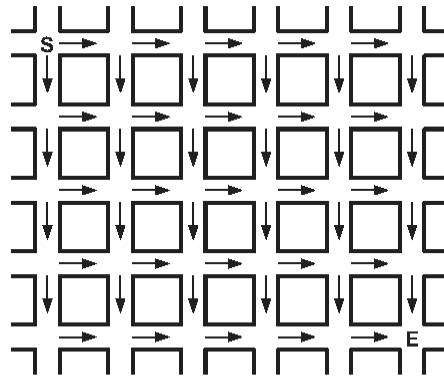


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

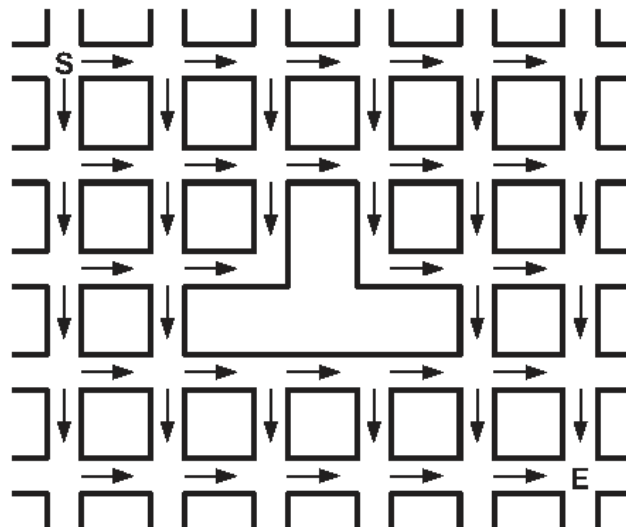


Figure B.



7494A063-1047-4D95-B349-7893D8760F0B

SPRING2014_CSCI570_EXAM2

#83 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

D14A7590-07A6-49CE-8C76-5CB6AC9F3506

SPRING2014_CSCI570_EXAM2

#83

9 of 10



Additional Space



E56842CD-1D21-4967-86DF-EA6B498B4F77

SPRING2014_CSCI570_EXAM2

#83 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



59E10CC8-6681-4FFA-BEB7-AF7230161BEC

SPRING2014_CSCI570_EXAM2

#84 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



9A436813-FA5C-4635-AD00-C24542A86986

SPRING2014_CSCI570_EXAM2

#84

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

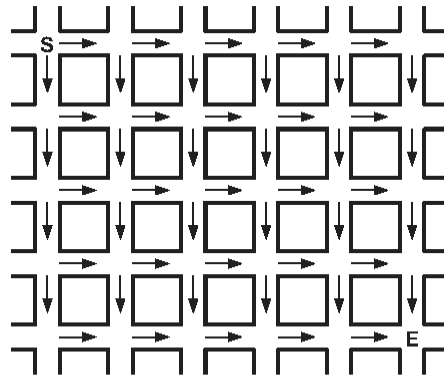


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



47F05340-86E0-49D0-B83F-D059C7A85903

SPRING2014_CSCI570_EXAM2

#84 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

C3A27A22-B72D-415D-8AFE-A6B7B3399E5E

SPRING2014_CSCI570_EXAM2

#84

9 of 10



Additional Space



064B3A52-70E3-4575-861B-68C20A9B8662

SPRING2014_CSCI570_EXAM2

#84 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



4643D353-1202-4499-B66E-5C2F72692DD2

SPRING2014_CSCI570_EXAM2

#85

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



B820FD64-E991-464F-B872-B9D4D09EFF96

SPRING2014_CSCI570_EXAM2

#85 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

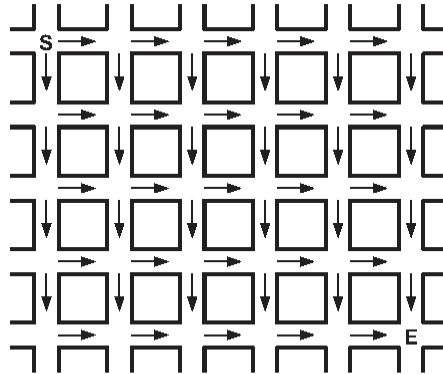


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

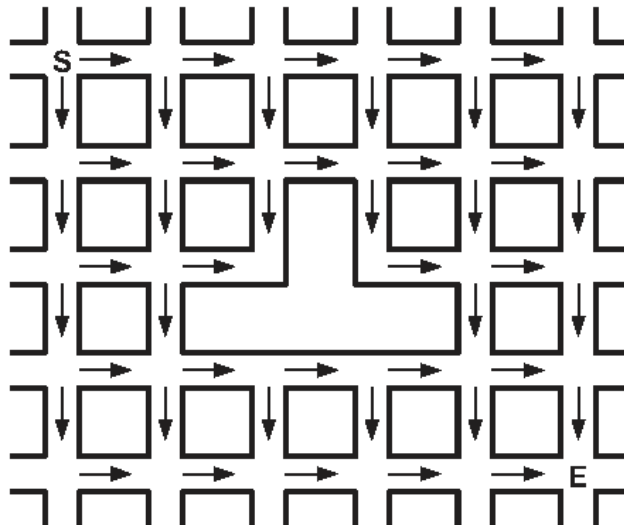


Figure B.



1397EE70-59CE-4B74-A487-F3671239BFE2

SPRING2014_CSCI570_EXAM2

#85 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

B32C3C72-E8B6-4758-8C41-BAD53289CF53

SPRING2014_CSCI570_EXAM2

#85

9 of 10



Additional Space



9920F1E6-6E39-4E2C-BD1D-6382D0A69FDC

SPRING2014_CSCI570_EXAM2

#85 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



47E4DBE0-6690-47AE-A8E4-B754DD89B727

SPRING2014_CSCI570_EXAM2

#86

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



150EB5E5-1FBB-402F-93F6-A2A1DB86B010

SPRING2014_CSCI570_EXAM2

#86

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

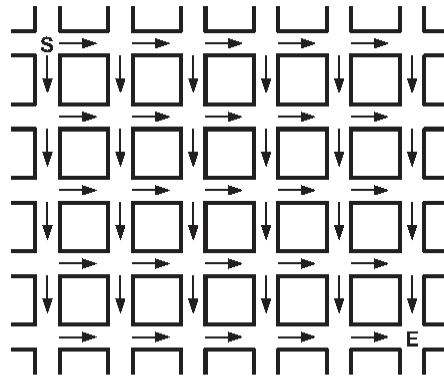


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

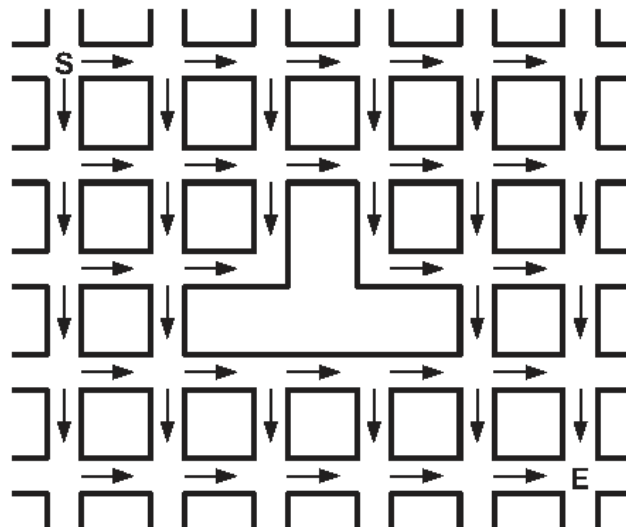


Figure B.



AA11A881-5D2A-4A72-9510-1CAC8008D67A

SPRING2014_CSCI570_EXAM2

#86 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

6BE7AECD-FD03-426D-9F34-439B854489F6

SPRING2014_CSCI570_EXAM2

#86 9 of 10



Additional Space



FC014284-7892-410E-95EF-D00B62ADB098

SPRING2014_CSCI570_EXAM2

#86 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



825BEE73-6178-405F-A10E-CAC1C286555E

SPRING2014_CSCI570_EXAM2

#87 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



0376EE32-9172-46D2-A4AE-0E2D31FB6C6B

SPRING2014_CSCI570_EXAM2

#87 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

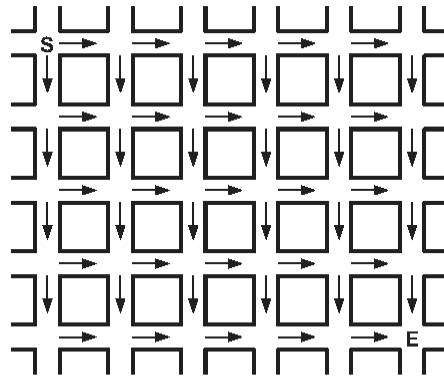


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



970295E1-1F71-4DFD-A54B-B87A71D2C944

SPRING2014_CSCI570_EXAM2

#87 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

FBC95B92-0668-45F1-95DE-B99F40C54659

SPRING2014_CSCI570_EXAM2

#87

9 of 10



Additional Space



462D4921-B262-432F-8B9B-23978919C05B

SPRING2014_CSCI570_EXAM2

#87 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



6BC4F6AB-8E91-42E8-B6A1-A52FD5192AB0

SPRING2014_CSCI570_EXAM2

#88

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

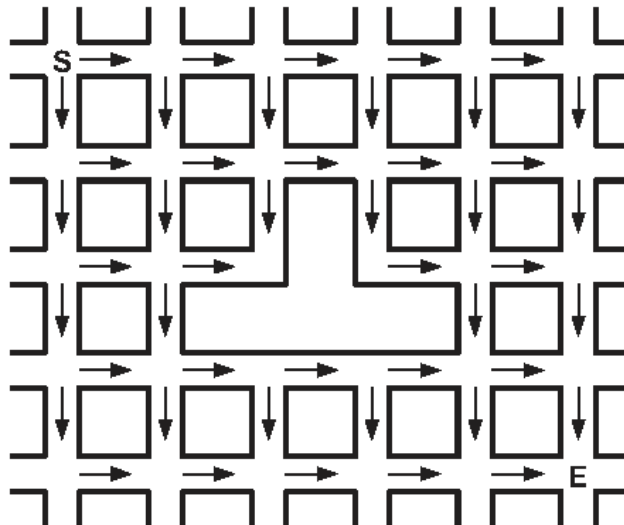


Figure B.



F02248E6-722F-4C15-BBDF-B3C7C4FDE0B1

SPRING2014_CSCI570_EXAM2

#88 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

02174E2D-F8AE-4A6A-9312-3D3AE113B06B

SPRING2014_CSCI570_EXAM2

#88

9 of 10



Additional Space



974EC5C8-2535-412D-9603-6A5D54365FA7

SPRING2014_CSCI570_EXAM2

#88 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



A8E22650-5F57-4E73-B6B1-54B0DF4C8116

SPRING2014_CSCI570_EXAM2

#89 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



A81C473E-4541-4CD7-9722-7BB0A6545F1A

SPRING2014_CSCI570_EXAM2

#89

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

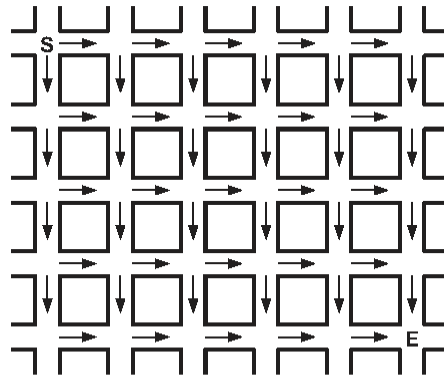


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



738C1295-ACF8-405F-9AD8-C8B0DFD8FB9D

SPRING2014_CSCI570_EXAM2

#89 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

D12CF203-F962-45E4-873F-0C2176BA2A41

SPRING2014_CSCI570_EXAM2

#89

9 of 10



Additional Space



A2DDFDA3-4564-4353-9EFF-D29DB2B2A028

SPRING2014_CSCI570_EXAM2

#89 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



28CAE18B-1F80-4E5F-B845-83C04484B7BB

SPRING2014_CSCI570_EXAM2

#90 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



9217AEEB-44F8-4245-ACAF-E4BA4379353D

SPRING2014_CSCI570_EXAM2

#90 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

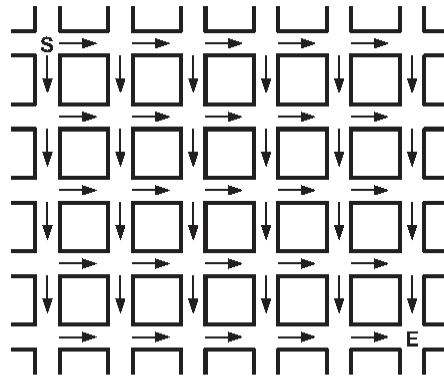


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

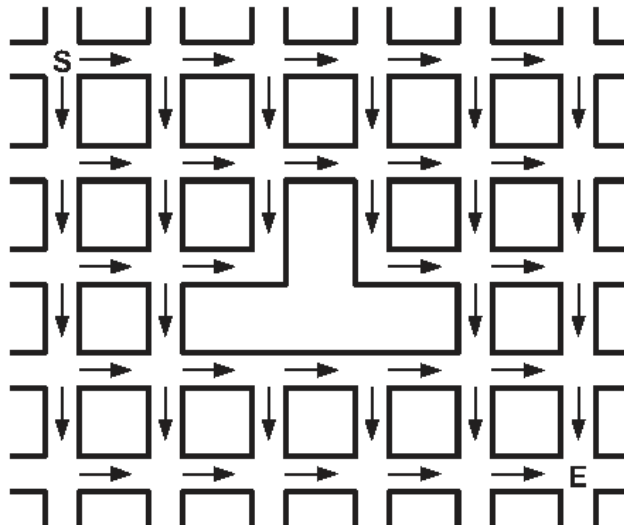


Figure B.



B27DA593-B535-47F0-8648-EA028A615F96

SPRING2014_CSCI570_EXAM2

#90 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

600A74FA-3EE3-4A74-AE84-4CB0F268CA2B

SPRING2014_CSCI570_EXAM2

#90 9 of 10



Additional Space



27889F4A-8F1C-443F-AF19-349B04F83F0C

SPRING2014_CSCI570_EXAM2

#90 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



6B07EDCE-060A-4CA4-89B3-7E7FFEDDBD83

SPRING2014_CSCI570_EXAM2

#91 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



F5B7B5EC-AD51-4EF5-9A69-4E026790A511

SPRING2014_CSCI570_EXAM2

#91 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

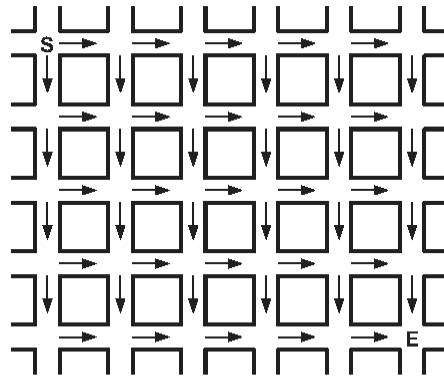


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



39777314-DD37-4B78-94E7-CE68A71FE596

SPRING2014_CSCI570_EXAM2

#91 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

3C59E6CA-1383-4B7C-B69F-7DAD02032E16

SPRING2014_CSCI570_EXAM2

#91 9 of 10



Additional Space



0374A7E4-E134-470A-ADFF-4739CB9BD3B0

SPRING2014_CSCI570_EXAM2

#91 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



4499A13F-2F78-4B42-BC3C-A2B10245F444

SPRING2014_CSCI570_EXAM2

#92 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



7316368F-7FF2-4287-A435-E2D81BF5E35A

SPRING2014_CSCI570_EXAM2

#92 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

94693D42-4D61-4567-A26B-EEF939053964

SPRING2014_CSCI570_EXAM2

#92 9 of 10



Additional Space



19C47F84-526D-4B9C-9B82-123FEB6428FC

SPRING2014_CSCI570_EXAM2

#92 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



B9649ECE-763C-4CCC-9344-9B69DFD2D1FF

SPRING2014_CSCI570_EXAM2

#93 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



987BB1AB-D873-4A1C-8ED6-3D8DCF6A85F6

SPRING2014_CSCI570_EXAM2

#93 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

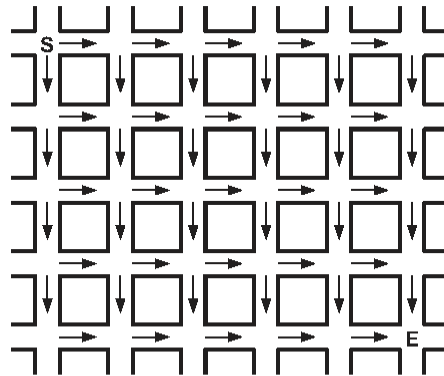


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



D86D39A0-5A88-4794-AC54-2D6AFD45DDF5

SPRING2014_CSCI570_EXAM2

#93 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

0B59A182-9CF7-43A2-B3F5-A8EDE1AAA79B

SPRING2014_CSCI570_EXAM2

#93 9 of 10



Additional Space



8EE848A6-D23B-42A3-BC8B-7AF8EABB2B28

SPRING2014_CSCI570_EXAM2

#93 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



0B87F412-0F0E-4F31-99EB-E14461494476

SPRING2014_CSCI570_EXAM2

#94 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



DD2FB515-40EC-4A62-9A6F-2AB8080C70D8

SPRING2014_CSCI570_EXAM2

#94 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

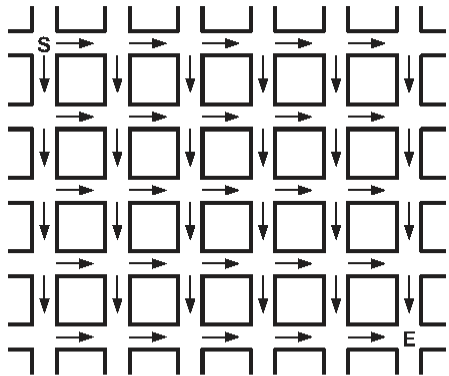


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



29A9A98D-6905-49D5-8775-E216A69AAD42

SPRING2014_CSCI570_EXAM2

#94 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

032200F2-EDFF-42C0-AF89-DC5F71D099D0

SPRING2014_CSCI570_EXAM2

#94 9 of 10



Additional Space



0ECEE51-6466-43A4-A4CA-ABCF2E24D0EF

SPRING2014_CSCI570_EXAM2

#94 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



3F8577EE-0872-48FE-A1A1-0C0110E44DA4

SPRING2014_CSCI570_EXAM2

#95 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



9A3CF3DA-9EA2-45CA-9C0E-8F9D1F557FAA

SPRING2014_CSCI570_EXAM2

#95

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

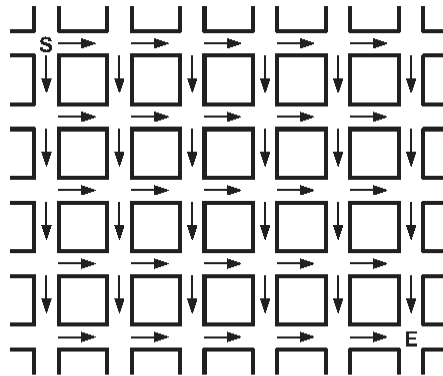


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



FFFBB61F-661F-46FE-8ED1-3EDFBEA92CCB

SPRING2014_CSCI570_EXAM2

#95 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

4C096E4D-85BA-4EB5-BFEC-04684C7D6F13

SPRING2014_CSCI570_EXAM2

#95

9 of 10



Additional Space



E4B5B79C-8514-476E-BD0F-882067C46D6B

SPRING2014_CSCI570_EXAM2

#95 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



EFBAACDD-1AA5-4675-88D3-5658FDC404CB

SPRING2014_CSCI570_EXAM2

#96 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

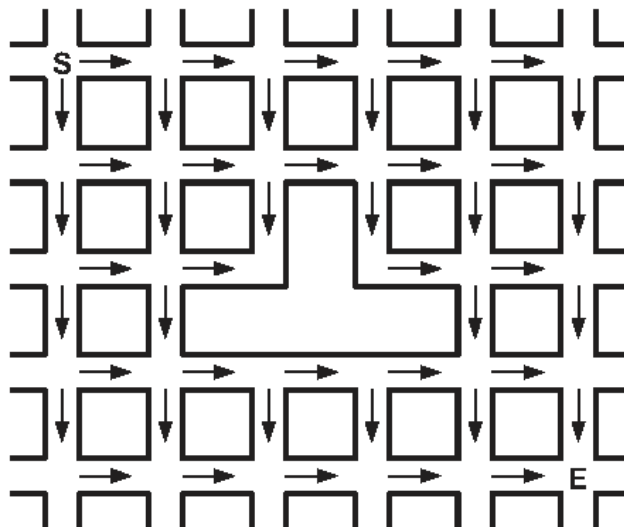


Figure B.



381715A2-7D43-4480-A4A9-D7EF1464A5CA

SPRING2014_CSCI570_EXAM2

#96 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

2FE62512-B4C5-48AA-8652-08C4C027176C

SPRING2014_CSCI570_EXAM2

#96

9 of 10



Additional Space



FF51200B-5E62-4441-825A-DADF6924488A

SPRING2014_CSCI570_EXAM2

#96 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



4558075F-9E6A-41FE-8F59-35D0AD7520DD

SPRING2014_CSCI570_EXAM2

#97 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

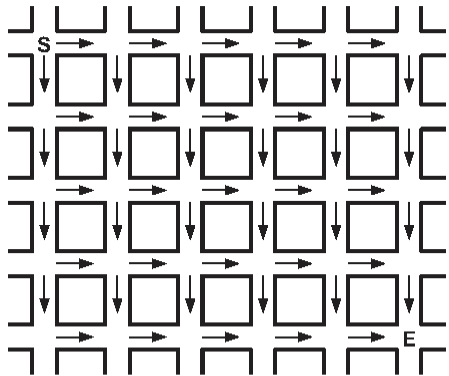


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



85C11491-A6C0-4FF9-8F0D-7B3BFE1F548C

SPRING2014_CSCI570_EXAM2

#97 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

0E4A1DD7-DF7C-4236-98A0-83B4BF0FE854

SPRING2014_CSCI570_EXAM2

#97

9 of 10



Additional Space



F359649E-9CA6-4146-B201-A55DAD801579

SPRING2014_CSCI570_EXAM2

#97 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



1981D849-D220-4607-A0F7-B27C73E8B264

SPRING2014_CSCI570_EXAM2

#98 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



E24D9538-C19C-48B8-B01E-4D8E6F76D1CF

SPRING2014_CSCI570_EXAM2

#98

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

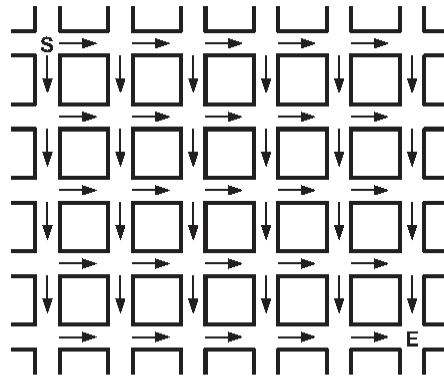


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



51EAEBA-5BD8-42E0-A3B3-03D7F6D614FB

SPRING2014_CSCI570_EXAM2

#98 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

39BBCC85-1C6B-46F6-BDD1-3E22380B6209

SPRING2014_CSCI570_EXAM2

#98

9 of 10



Additional Space



30E5AEE9-21F1-434A-BFD3-C17C3FFA4413

SPRING2014_CSCI570_EXAM2

#98 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



0122A02C-2B3A-4CC0-80D1-C0EEE5021EE7

SPRING2014_CSCI570_EXAM2

#99 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



61EC8CA0-3A3A-41A6-9D7E-13F3F2E5ED0F

SPRING2014_CSCI570_EXAM2

#99

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

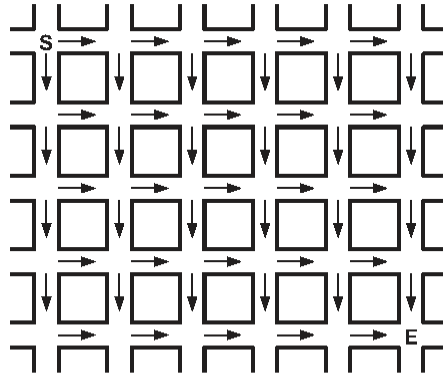


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



63BDC221-C463-4C6E-8EFD-F463B2C52234

SPRING2014_CSCI570_EXAM2

#99 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

84F7E7AC-3F5E-4292-95AD-5D44B6262657

SPRING2014_CSCI570_EXAM2

#99

9 of 10



Additional Space



8158C40E-7B44-4C5D-B13E-F15BD68680A9

SPRING2014_CSCI570_EXAM2

#99 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



E0DD3EEF-4E45-4DE0-95B7-1CF07163683A

SPRING2014_CSCI570_EXAM2

#100 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



585BE0E3-BFE3-4D8A-A85E-A0237BE1E442

SPRING2014_CSCI570_EXAM2

#100 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

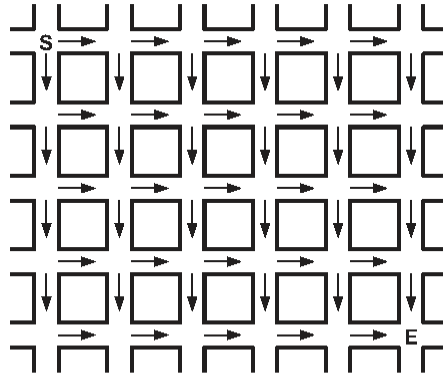


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



45034ABA-C75F-4636-ADD8-C098A625411D

SPRING2014_CSCI570_EXAM2

#100 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

E9AE4173-12C8-4C6F-96FA-63956B714A57

SPRING2014_CSCI570_EXAM2

#100

9 of 10



Additional Space



B071D99D-E14A-4E97-BF2E-371BDB678853

SPRING2014_CSCI570_EXAM2

#100 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



8D51820D-C196-4119-87CA-63C9FF24A1E1

SPRING2014_CSCI570_EXAM2

#101 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



1C5C858C-8866-46E4-AD1E-852B98D80857

SPRING2014_CSCI570_EXAM2

#101 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

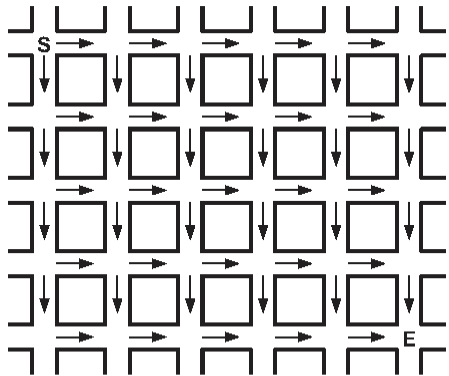


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

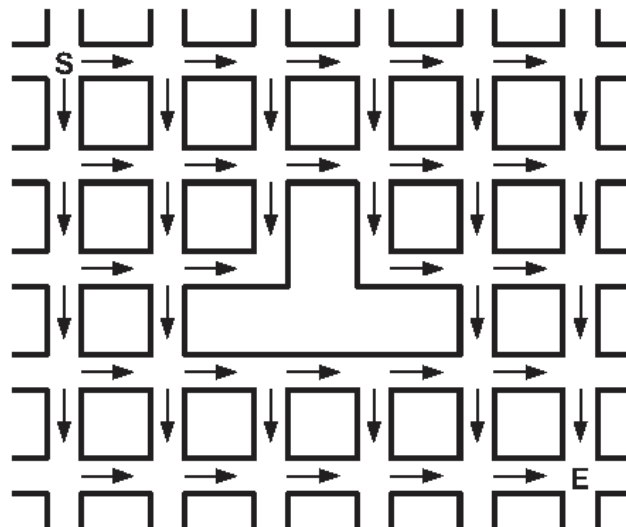


Figure B.



15A9FA5B-7311-414C-98CA-721D0B028C1C

SPRING2014_CSCI570_EXAM2

#101 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

878C2C79-D145-409C-A149-FC015517BC4F

SPRING2014_CSCI570_EXAM2

#101

9 of 10



Additional Space



63F93A54-3A07-454A-829E-439EADD224F4

SPRING2014_CSCI570_EXAM2

#101 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



73AF32E2-FBE0-4485-BD2B-E62CC3497482

SPRING2014_CSCI570_EXAM2

#102 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

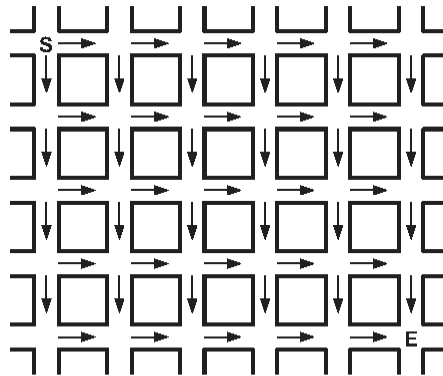


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



AB4BE53D-806E-4733-8717-3AF31ABD23FA

SPRING2014_CSCI570_EXAM2

#102 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9C115B3B-C3C2-4AB6-A59C-DB6B19226333

SPRING2014_CSCI570_EXAM2

#102

9 of 10



Additional Space



8BCD0CD6-B8F0-4B20-92B0-FB5204BF1351

SPRING2014_CSCI570_EXAM2

#102 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



2E8AE701-5495-467B-A55C-4D6EEE7B5BC0

SPRING2014_CSCI570_EXAM2

#103 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



9A92E7BD-8413-426E-8A69-8587D118493B

SPRING2014_CSCI570_EXAM2

#103

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

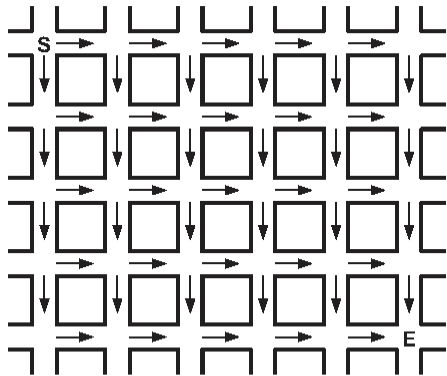


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



7F99CA5F-C80A-4356-B748-A6343470860F

SPRING2014_CSCI570_EXAM2

#103 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

707BD640-E2B0-45A5-A307-5289FC7FEE70

SPRING2014_CSCI570_EXAM2

#103

9 of 10



Additional Space



F39D8F81-04F6-495E-9EF7-5562BFF003AB

SPRING2014_CSCI570_EXAM2

#103 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



F6E1250F-06BC-4EB3-AF8D-614420B25524

SPRING2014_CSCI570_EXAM2

#104 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



6AF8ADBF-6C51-43D5-924C-8F9FC458FB03

SPRING2014_CSCI570_EXAM2

#104 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



2E3F12D4-1596-4849-A215-94FEF4D9DAB5

SPRING2014_CSCI570_EXAM2

#104 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

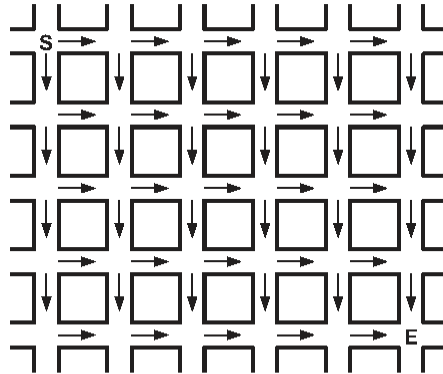


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

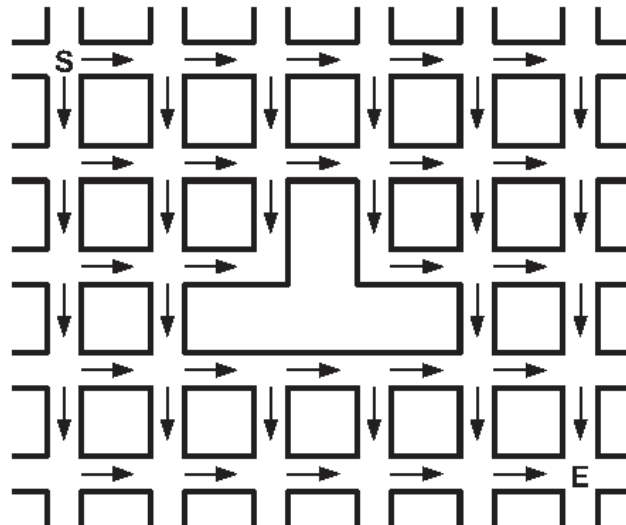


Figure B.



8475080D-3274-4E8C-8BE2-6CF942D763DC

SPRING2014_CSCI570_EXAM2

#104 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

B9B2D042-5DF6-45D8-BFFF-0AC169CB2FE4

SPRING2014_CSCI570_EXAM2

#104 9 of 10



Additional Space



0628C648-2EEE-4193-9969-D97767D72B08

SPRING2014_CSCI570_EXAM2

#104 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



3A1B4732-48DC-49CF-99FC-C63B3E9271B8

SPRING2014_CSCI570_EXAM2

#105 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



E843CFB1-FECA-4726-9951-63F11A2F1FB7

SPRING2014_CSCI570_EXAM2

#105 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



A90042F5-0A24-40C6-88A6-E296B57570B3

SPRING2014_CSCI570_EXAM2

#105 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

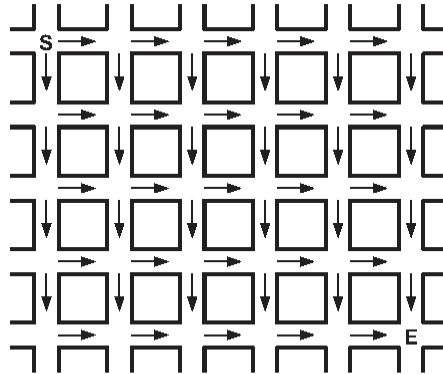


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



06B74B5A-7537-4AB0-A145-E4B411D94E1D

SPRING2014_CSCI570_EXAM2

#105 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9601FFE9-F4E1-4CDD-BB23-6B10C94D7AAD

SPRING2014_CSCI570_EXAM2

#105

9 of 10



Additional Space



6C269B6B-3488-4691-9D49-980CDB8E21B8

SPRING2014_CSCI570_EXAM2

#105 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



DB106D28-F28E-4E63-B255-3FE9362E4928

SPRING2014_CSCI570_EXAM2

#106 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



6FC35D1C-5A04-494C-B186-150A77EF1A3A

SPRING2014_CSCI570_EXAM2

#106

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

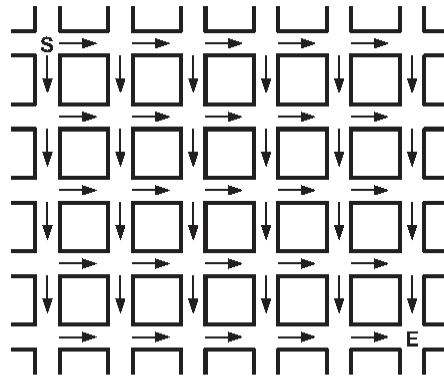


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



FA5D321A-FCE0-481A-A994-7BD70952A962

SPRING2014_CSCI570_EXAM2

#106 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

A94F824B-9B07-47FD-A950-7413EA574A3B

SPRING2014_CSCI570_EXAM2

#106

9 of 10



Additional Space



E80416DD-D8CB-4B9F-A21A-AA9D1A39C3B3

SPRING2014_CSCI570_EXAM2

#106 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



44033C74-FA24-4EA9-A824-FBAF3B50F6C2

SPRING2014_CSCI570_EXAM2

#107 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



B462BC66-7E68-47E3-A90C-5B4E7AB58B56

SPRING2014_CSCI570_EXAM2

#107 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

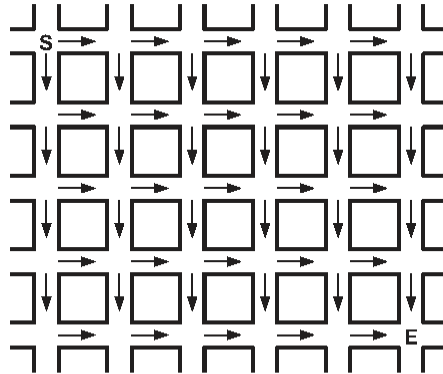


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

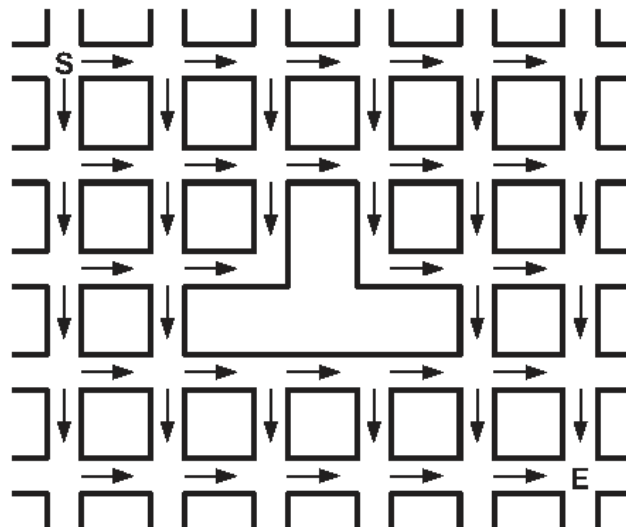


Figure B.



6F9D19D0-8D32-47A6-B80E-4A4A91526F75

SPRING2014_CSCI570_EXAM2

#107 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

3BFF1522-B080-4A32-AB47-1A3C46145B3D

SPRING2014_CSCI570_EXAM2

#107

9 of 10



Additional Space



A41BC3F2-0F02-41A7-B551-00D7002C69CD

SPRING2014_CSCI570_EXAM2

#107 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



58C790DF-492E-48A5-8B99-7EB55CAAC3C8

SPRING2014_CSCI570_EXAM2

#108

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



E7E56C99-3FE7-4D09-BEF3-0551EE7FD08B

SPRING2014_CSCI570_EXAM2

#108 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

CA20DDB4-2753-4656-BEDC-9A1C894F5443

SPRING2014_CSCI570_EXAM2

#108

9 of 10



Additional Space



09F6DED5-0BF4-40AD-B384-20AE1B48DB52

SPRING2014_CSCI570_EXAM2

#108 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



DE3CBB9F-46F8-4E99-AA18-454EB3EA1A90

SPRING2014_CSCI570_EXAM2

#109 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

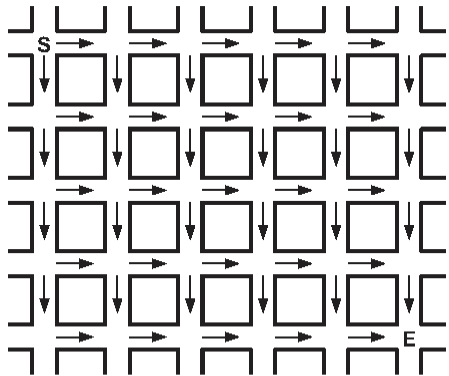


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

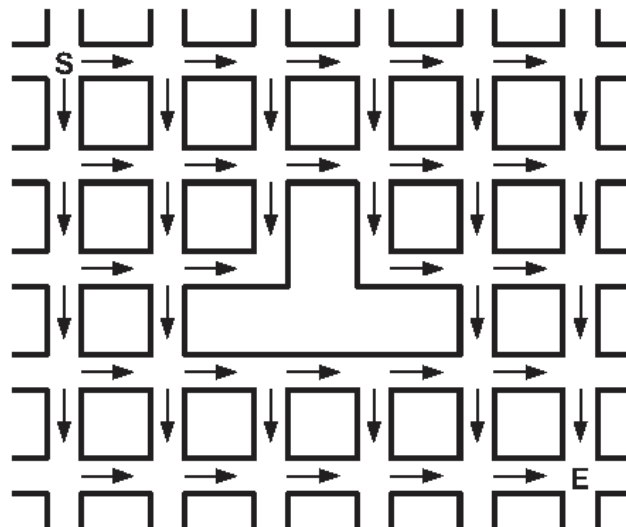


Figure B.



F87348A2-9219-4131-B428-E63650439047

SPRING2014_CSCI570_EXAM2

#109 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

A9F6A4BC-026E-47E6-8944-D2927D2BF1E4

SPRING2014_CSCI570_EXAM2

#109

9 of 10



Additional Space



29798289-1F04-43F0-BA12-8324487C6571

SPRING2014_CSCI570_EXAM2

#109 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



24545850-F6C9-4112-A7F9-CE5002F238E9

SPRING2014_CSCI570_EXAM2

#110 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



E63C9583-F06B-4A98-BC45-7AE7CA1B2A59

SPRING2014_CSCI570_EXAM2

#110 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



971E2759-4017-4445-9C9F-240B55D284CD

SPRING2014_CSCI570_EXAM2

#110

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

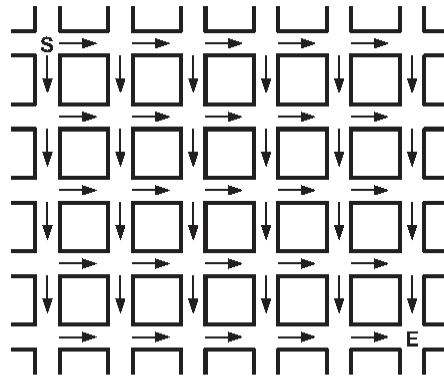


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

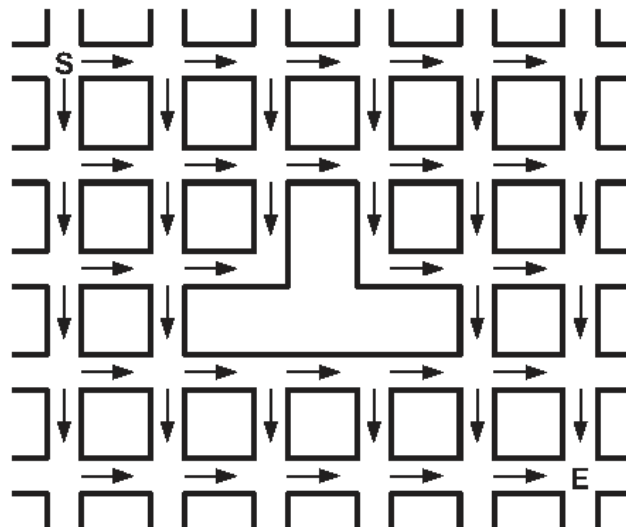


Figure B.



40B53096-F3F0-416A-91AB-4848F3F29513

SPRING2014_CSCI570_EXAM2

#110 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

F0675689-A142-46F9-B65C-2EA645000BAB

SPRING2014_CSCI570_EXAM2

#110

9 of 10



Additional Space



6A14B91E-2B1A-4854-B9C9-E23C33CED9B8

SPRING2014_CSCI570_EXAM2

#110 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



5AAB9880-452F-4610-B671-1CE38CEC1B22

SPRING2014_CSCI570_EXAM2

#111 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



05349C28-D92E-45E5-B07E-9525358A3F69

SPRING2014_CSCI570_EXAM2

#111

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

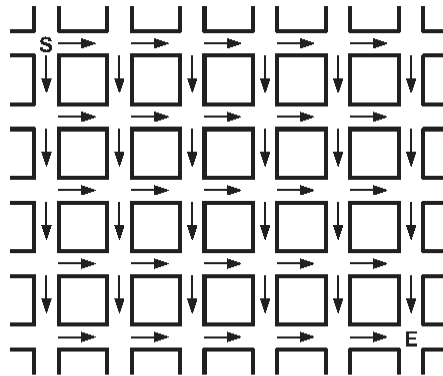


Figure A.

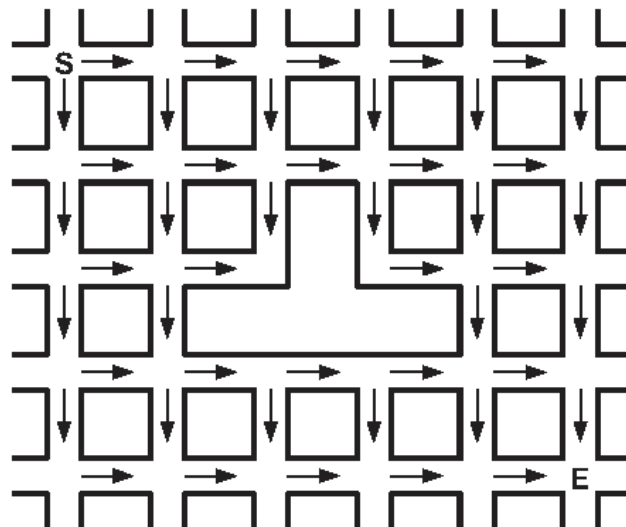


Figure B.



04C2D22F-1BC9-45EF-A081-5D7A2A2E662E

SPRING2014_CSCI570_EXAM2

#111 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

08D83E66-4239-4ED4-B863-089019315B97

SPRING2014_CSCI570_EXAM2

#111

9 of 10



Additional Space



74683FD8-64C0-45FB-9A69-6A17EC5D1260

SPRING2014_CSCI570_EXAM2

#111 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



2BCE188B-6303-4BA9-A269-0E5B7AFE5309

SPRING2014_CSCI570_EXAM2

#112 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

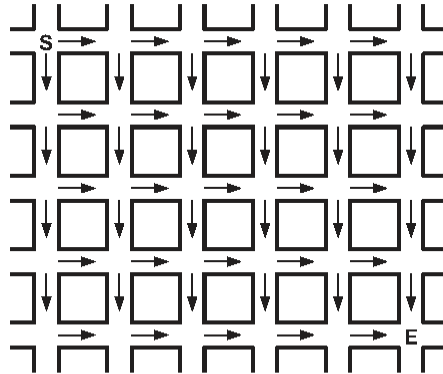


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



0E667A86-4680-4798-8777-F62E89FEA2E1

SPRING2014_CSCI570_EXAM2

#112 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

ECED8A8F-5B89-4730-9E5C-59F89BA49CA6

SPRING2014_CSCI570_EXAM2

#112

9 of 10



Additional Space



8012B9FC-BCDB-4D1A-9318-CF1843B88BA2

SPRING2014_CSCI570_EXAM2

#112 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



344266B4-4845-463D-A73C-06BBB5865C3F

SPRING2014_CSCI570_EXAM2

#113

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

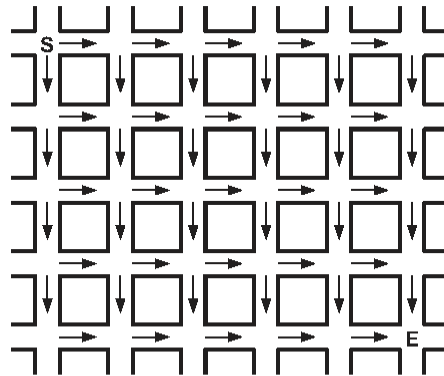


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



4E318548-BA68-4C2D-B5E6-05F9ECA5AE51

SPRING2014_CSCI570_EXAM2

#113 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9DE4091C-FD44-48EC-9BBA-DEB3791D0747

SPRING2014_CSCI570_EXAM2

#113

9 of 10



Additional Space



EF000E51-D5BB-45CA-9D45-CC6B82F83C9F

SPRING2014_CSCI570_EXAM2

#113 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



11BC71FC-0500-40EA-BE81-F233C2FD2D52

SPRING2014_CSCI570_EXAM2

#114 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



8B1395D9-453A-40B2-9345-C230FF757CB2

SPRING2014_CSCI570_EXAM2

#114 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

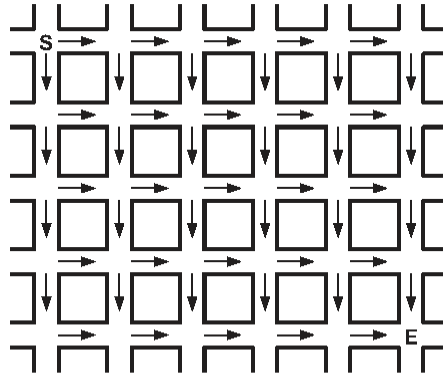


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

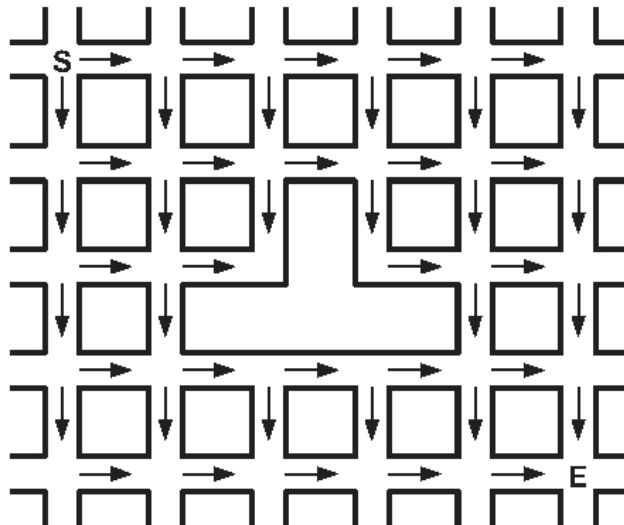


Figure B.



1D28B31A-1669-41E5-9F54-2BD1C2F945DD

SPRING2014_CSCI570_EXAM2

#114 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

812C9F8D-9E00-474A-A35B-F2B584144346

SPRING2014_CSCI570_EXAM2

#114

9 of 10



Additional Space



CD19CC9C-4056-4F01-A56F-852F20155FA3

SPRING2014_CSCI570_EXAM2

#114 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



64D5FBCE-3E00-4425-B400-EF994C9D5413

SPRING2014_CSCI570_EXAM2

#115 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



9554EA50-3A85-47D8-AB6A-54575E251CAE

SPRING2014_CSCI570_EXAM2

#115 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

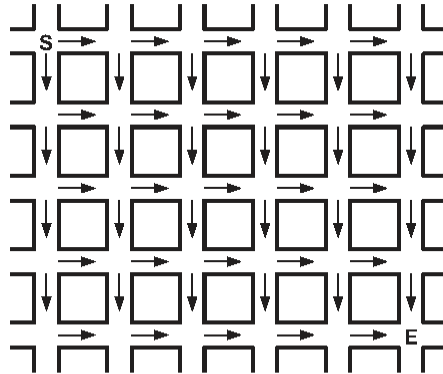


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

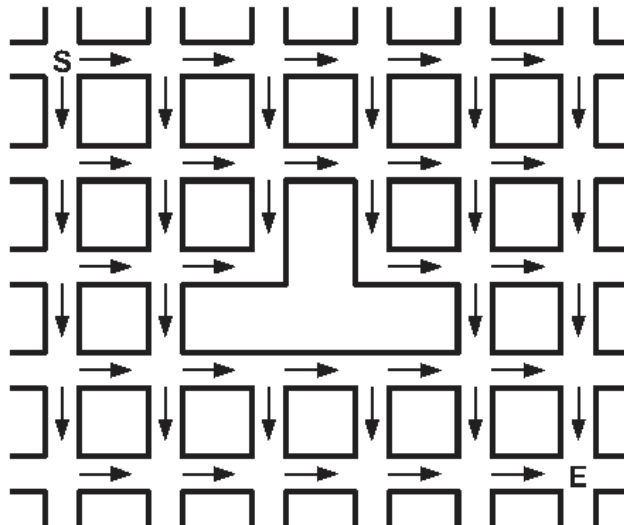


Figure B.



C8B82042-E21D-4E11-8CF2-F004A740D864

SPRING2014_CSCI570_EXAM2

#115 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

47A94D6F-5DE6-4F6A-91AE-8186F6CECF68

SPRING2014_CSCI570_EXAM2

#115

9 of 10



Additional Space



B3BD68F7-A5B7-4B4C-AE94-4CCC39DFF00A

SPRING2014_CSCI570_EXAM2

#115 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



155936C8-E1B4-4A8C-BD21-1F7A16CF47D0

SPRING2014_CSCI570_EXAM2

#116 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



A6D788DD-EE26-4136-9CE8-EC627B04A537

SPRING2014_CSCI570_EXAM2

#116 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



F34FA74C-DD25-479B-AA0B-74D167E3941A

SPRING2014_CSCI570_EXAM2

#116

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

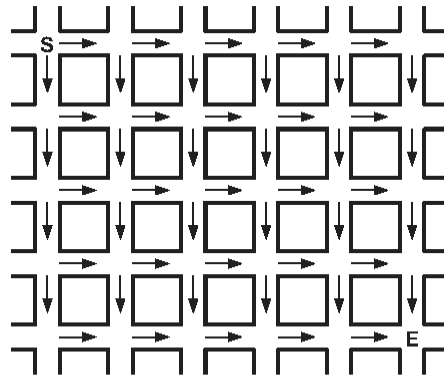


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

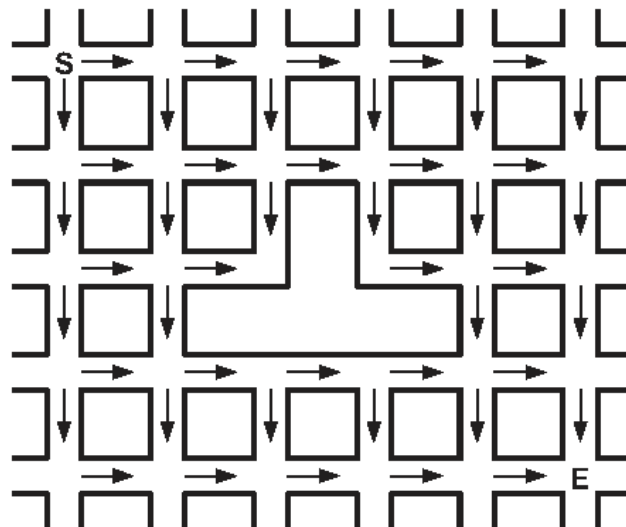


Figure B.



27533162-4442-45F4-B7D6-76E510C530B4

SPRING2014_CSCI570_EXAM2

#116 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

F82F75D8-753D-4495-92A9-6A43E6F6A69C

SPRING2014_CSCI570_EXAM2

#116

9 of 10



Additional Space



E3EE48EE-A0CA-4517-B529-B99DCE373098

SPRING2014_CSCI570_EXAM2

#116 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



8F7EE4A9-CDAB-4E0D-9E12-9E484CB7A237

SPRING2014_CSCI570_EXAM2

#117 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



466E7856-7B98-4C74-96CC-12AE067825D6

SPRING2014_CSCI570_EXAM2

#117 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

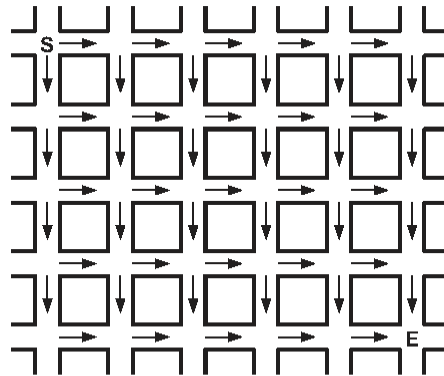


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



004D34DE-460F-40EC-9AE7-1E10AB8A7D45

SPRING2014_CSCI570_EXAM2

#117 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

6BB8B3AD-8D20-4210-8166-DB3DDBD5159D

SPRING2014_CSCI570_EXAM2

#117

9 of 10



Additional Space



FA9C26B0-C1F2-47A4-9405-B49E269E8BCB

SPRING2014_CSCI570_EXAM2

#117 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1D6ACA3B-349D-4304-92C0-520828E8C442

SPRING2014_CSCI570_EXAM2

#118 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



D5ACEDB5-DF59-468C-8258-57CC6F8735E5

SPRING2014_CSCI570_EXAM2

#118

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

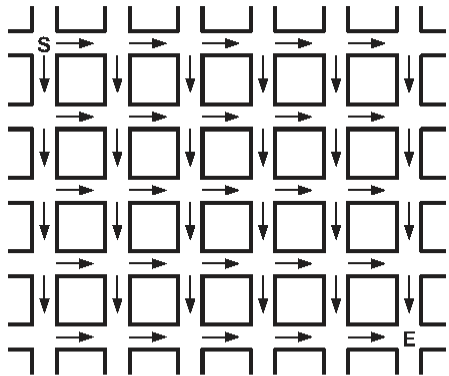


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



8B86FFDD-A86C-4066-8616-F7C041439256

SPRING2014_CSCI570_EXAM2

#118 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9E46DA87-D58C-4759-A1A3-AEDBFBBF10CF

SPRING2014_CSCI570_EXAM2

#118

9 of 10



Additional Space



7B49BCDE-E874-41EB-AFF2-37A2A4638558

SPRING2014_CSCI570_EXAM2

#118 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



748ACBD5-9945-4163-9329-DD39D495D8BB

SPRING2014_CSCI570_EXAM2

#119

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

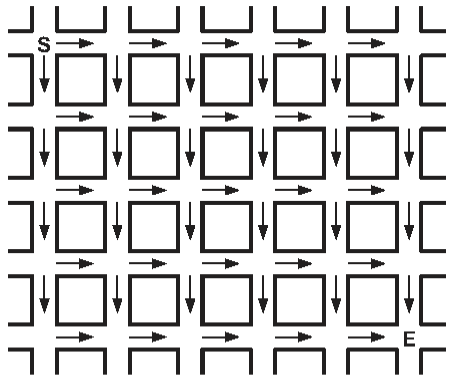


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

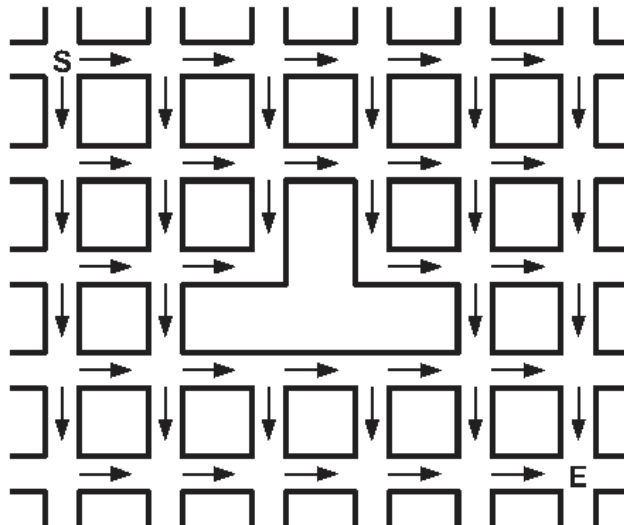


Figure B.



0D00069B-4295-4159-BE88-A6E806BF51F6

SPRING2014_CSCI570_EXAM2

#119 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

6C70B2C9-9B9A-458C-80E6-6012CC1787D2

SPRING2014_CSCI570_EXAM2

#119

9 of 10



Additional Space



44594E03-072C-45AA-945E-257D8D5977B3

SPRING2014_CSCI570_EXAM2

#119 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



6C001257-0645-4DF0-A843-3B9725DC5AD1

SPRING2014_CSCI570_EXAM2

#120 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

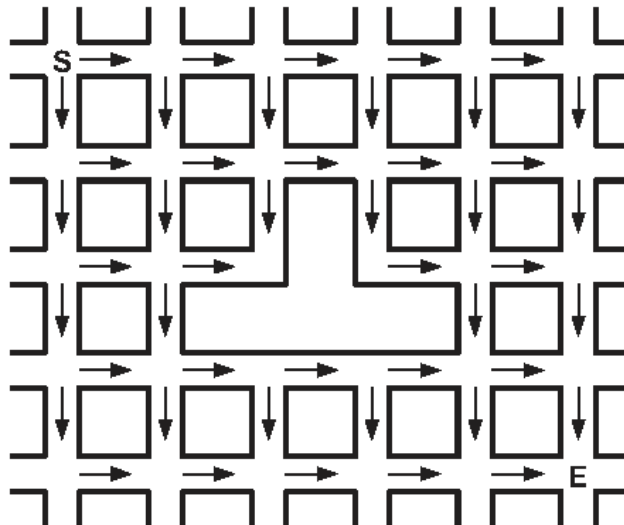


Figure B.



8A0F03D1-DD35-4461-9CC3-087FB473AFC4

SPRING2014_CSCI570_EXAM2

#120 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

7C087115-42A3-45A5-AA62-F72343F531F6

SPRING2014_CSCI570_EXAM2

#120

9 of 10



Additional Space



41590549-5D0C-4ACA-8D11-62EB53C11450

SPRING2014_CSCI570_EXAM2

#120 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



508FD0FE-14AD-4C92-B5DF-1E00169548D7

SPRING2014_CSCI570_EXAM2

#121 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



F5ABBA3C-9C32-4FF0-BAD0-4A3C0C374343

SPRING2014_CSCI570_EXAM2

#121 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

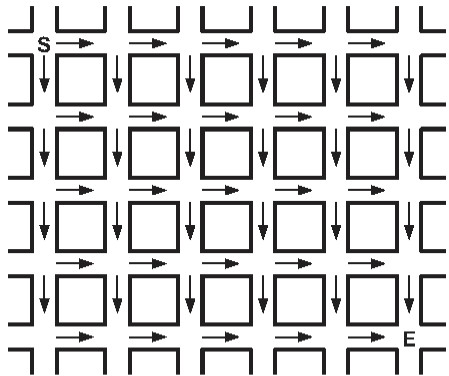


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



B3E3BAE2-BF80-4280-BB51-A68B99681B47

SPRING2014_CSCI570_EXAM2

#121 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9A984671-074F-4CA7-B7FC-141133EDA548

SPRING2014_CSCI570_EXAM2

#121

9 of 10



Additional Space



F2EC7D36-5000-4FEE-97A2-12DC43331FE2

SPRING2014_CSCI570_EXAM2

#121 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



EB5C711B-C8ED-48F9-AAAC-0783E6086269

SPRING2014_CSCI570_EXAM2

#122 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

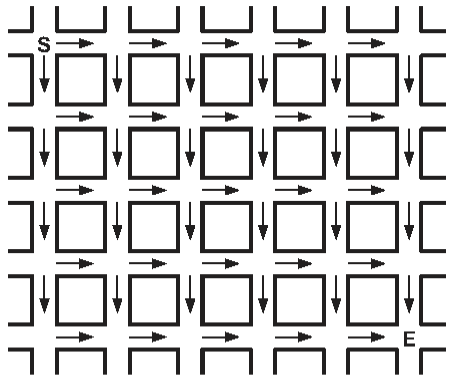


Figure A.



5BAF7B0F-B510-49BD-B6CB-C7F553BF7F0A

SPRING2014_CSCI570_EXAM2

#122 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

21C7A5ED-30CE-4012-A4FF-8DE8A5ACCAB1

SPRING2014_CSCI570_EXAM2

#122

9 of 10



Additional Space



FC81B01C-F7A9-4A6B-9CC7-6D3BD4674B28

SPRING2014_CSCI570_EXAM2

#122 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



C40AA1A4-45A0-4910-A1E4-DB9BC97ED604

SPRING2014_CSCI570_EXAM2

#123 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



A70B5880-3D60-42F3-A8C7-E46B11CC20D6

SPRING2014_CSCI570_EXAM2

#123

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

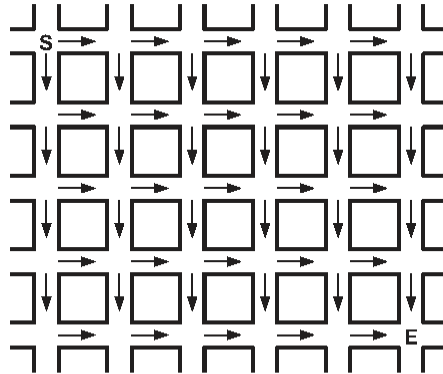


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



A69ABB11-610E-4284-A94A-6AE32288A8BE

SPRING2014_CSCI570_EXAM2

#123 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

93386928-1357-4DD2-A546-EA48EDB47D7D

SPRING2014_CSCI570_EXAM2

#123

9 of 10



Additional Space



7DE99D90-035F-4A8E-BAF5-7E2738B4C7C7

SPRING2014_CSCI570_EXAM2

#123 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



107999E0-794B-4D5E-BAEB-CF106B41AFF9

SPRING2014_CSCI570_EXAM2

#124 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



45854135-87E9-41FB-B7C0-CC94466FC1C7

SPRING2014_CSCI570_EXAM2

#124 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

B8314CC0-04D5-4EC3-8BA2-3DDE9B310EC4

SPRING2014_CSCI570_EXAM2

#124

9 of 10



Additional Space



83404406-0290-4433-8304-2E793E207511

SPRING2014_CSCI570_EXAM2

#124 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



6E753040-F4E5-4C52-A64C-02C73FC7BDAC

SPRING2014_CSCI570_EXAM2

#125 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

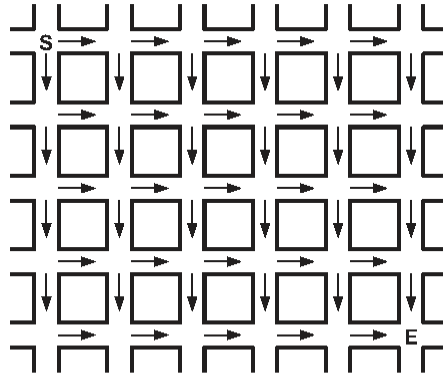


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



CC90C406-3667-4402-8AB8-BB96FA36C83E

SPRING2014_CSCI570_EXAM2

#125 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

1380A4A9-FD7D-4A48-911F-5FD813F8EDE7

SPRING2014_CSCI570_EXAM2

#125

9 of 10



Additional Space



5105C7DF-A3A4-4ED1-A65C-0C9F7744F83C

SPRING2014_CSCI570_EXAM2

#125 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



E34C2278-24B2-4CE2-84E0-74B5E3B8FF22

SPRING2014_CSCI570_EXAM2

#126 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



680809E1-D5DB-4EBC-8175-242461939350

SPRING2014_CSCI570_EXAM2

#126

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

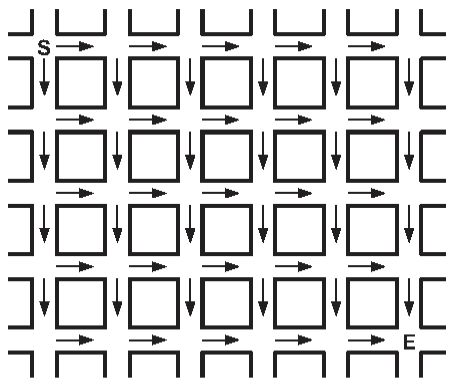


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



36DA8FF1-78E3-49D4-BCC3-13C114941356

SPRING2014_CSCI570_EXAM2

#126 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

A7913AD7-A4F9-435B-9165-6F54B4125D95

SPRING2014_CSCI570_EXAM2

#126

9 of 10



Additional Space



ED86154D-2841-459C-9B1C-CB1D3D561DB7

SPRING2014_CSCI570_EXAM2

#126 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



5465E930-9233-49EF-822C-E8F49ADD1E6C

SPRING2014_CSCI570_EXAM2

#127 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

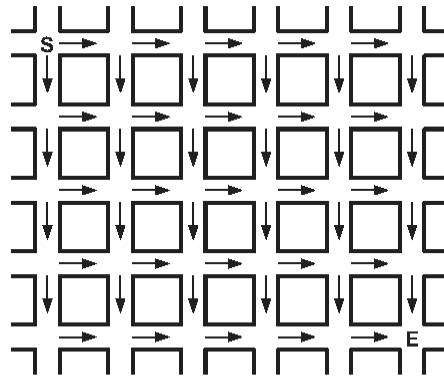


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

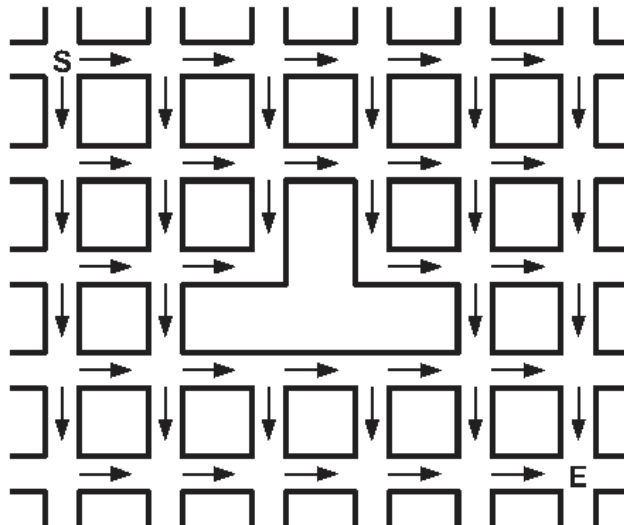


Figure B.



798546E7-E081-49E7-99A8-9E7E134E5887

SPRING2014_CSCI570_EXAM2

#127 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

F0722DDD-558D-46CF-A7C9-36C6029D35E7

SPRING2014_CSCI570_EXAM2

#127

9 of 10



Additional Space



BBCFA3EA-E71B-4C7C-B7FB-C6D2452FC78A

SPRING2014_CSCI570_EXAM2

#127 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



46E85B1E-20E0-48B1-83EE-83CE7ADEA3C7

SPRING2014_CSCI570_EXAM2

#128

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

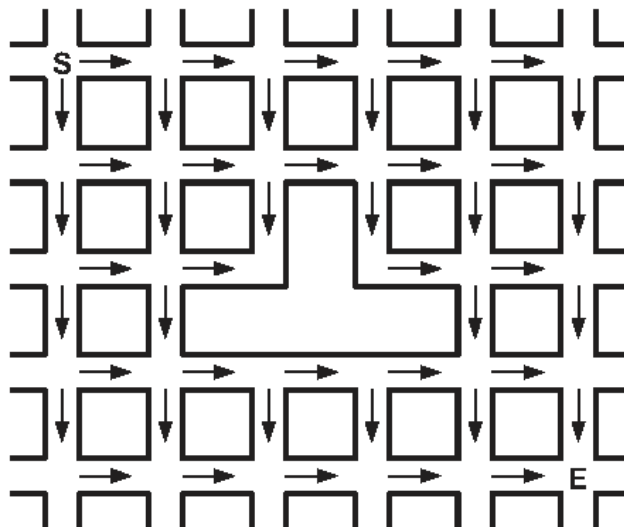


Figure B.



02CD896B-EE88-43EC-ACE6-C29E80E62CD5

SPRING2014_CSCI570_EXAM2

#128 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

D5EE7DF7-6A53-4B78-AE49-397B5F183CD0

SPRING2014_CSCI570_EXAM2

#128

9 of 10



Additional Space



E02DB813-F614-488E-AA8D-EA6C40D84B47

SPRING2014_CSCI570_EXAM2

#128 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



3FD7E83E-71E0-4F25-A557-284EA2F89F78

SPRING2014_CSCI570_EXAM2

#129 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



98F0C2C7-9A0F-42CA-B6F9-CBF32FBED846

SPRING2014_CSCI570_EXAM2

#129 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

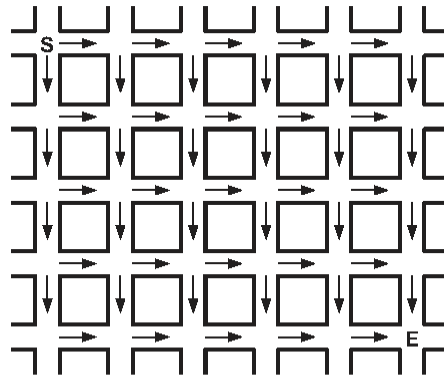


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

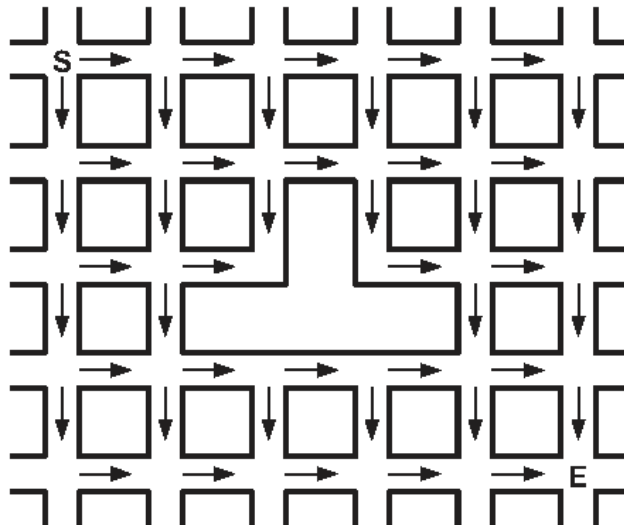


Figure B.



8916FBAC-B6FB-4BC6-BBDA-0C881AB06118

SPRING2014_CSCI570_EXAM2

#129 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

C156F69E-355A-457D-B7C3-A2BE65FB8F58

SPRING2014_CSCI570_EXAM2

#129

9 of 10



Additional Space



22136989-DDAC-4A40-A7BD-43747252825D

SPRING2014_CSCI570_EXAM2

#129 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



3EC53029-8C7D-46F6-B89B-D4177F828B87

SPRING2014_CSCI570_EXAM2

#130 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



BE10C6FD-CCE3-42E7-B168-C498DE93344A

SPRING2014_CSCI570_EXAM2

#130 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

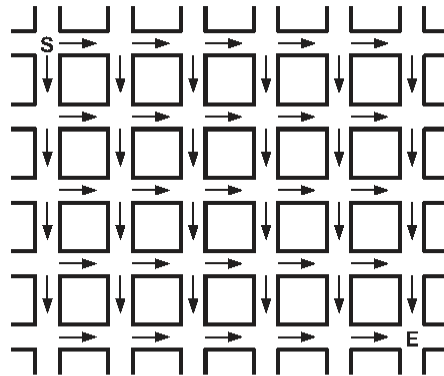


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



FD135F5F-6DBA-464B-8891-E7C1787E7C36

SPRING2014_CSCI570_EXAM2

#130 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

B311627E-9A22-4C49-8FAC-17AD32F7463A

SPRING2014_CSCI570_EXAM2

#130

9 of 10



Additional Space



6196623A-5FA4-4275-BB9C-8534406FD7B6

SPRING2014_CSCI570_EXAM2

#130 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



F4517BFE-7DCA-4958-8FAA-73C06E520696

SPRING2014_CSCI570_EXAM2

#131 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



42725A80-4DAA-468A-A896-91DAA799EC72

SPRING2014_CSCI570_EXAM2

#131 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

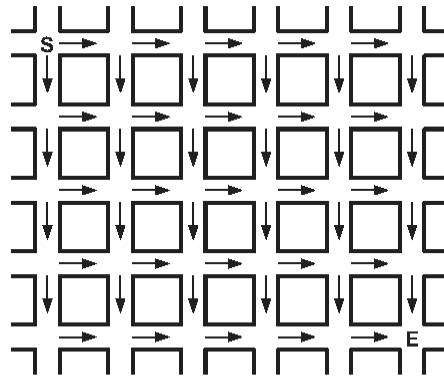


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



24F5E693-62E3-4603-B824-13089632C170

SPRING2014_CSCI570_EXAM2

#131 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

5E5D0648-16F7-4449-A782-2648A8FF3708

SPRING2014_CSCI570_EXAM2

#131

9 of 10



Additional Space



906C64A8-9394-40F4-868D-17F7FA592617

SPRING2014_CSCI570_EXAM2

#131 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



F5140758-74D5-44BC-9977-908DDFF91254

SPRING2014_CSCI570_EXAM2

#132 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



5) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

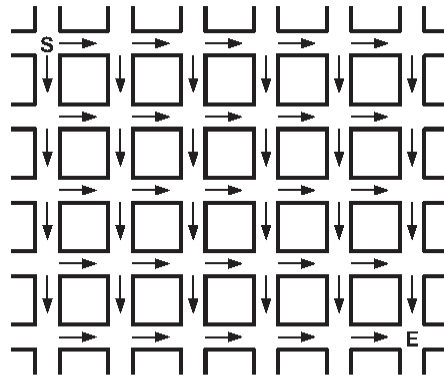


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

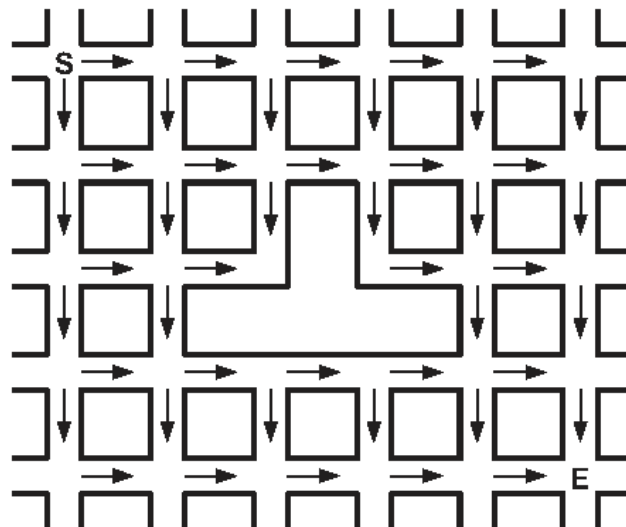


Figure B.



46081C1E-178C-4864-9C11-2E036B5CC064

SPRING2014_CSCI570_EXAM2

#132 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

E8AEA937-1572-422E-A83D-B9E160B06413

SPRING2014_CSCI570_EXAM2

#132 9 of 10



Additional Space



D0B6F9AC-EDA2-42E5-AC4F-6981145F1D34

SPRING2014_CSCI570_EXAM2

#132 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



F8BBB07F-2EA9-4D79-8E29-61745F2D1A86

SPRING2014_CSCI570_EXAM2

#133 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

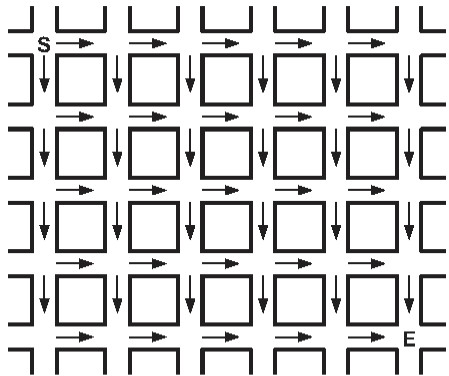


Figure A.



5166F9FA-F576-4E8C-98FD-FFE48969A187

SPRING2014_CSCI570_EXAM2

#133 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9BF1F009-C3B8-4751-B998-065C82073F8E

SPRING2014_CSCI570_EXAM2

#133

9 of 10



Additional Space



16400DA8-45E8-4D38-B013-D6C39D269C3D

SPRING2014_CSCI570_EXAM2

#133 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



4BAE7F87-AACF-4BF8-AF21-A341CAB73D91

SPRING2014_CSCI570_EXAM2

#134 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



5E488B54-DC8C-48AC-84BC-FE7E10564D70

SPRING2014_CSCI570_EXAM2

#134 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

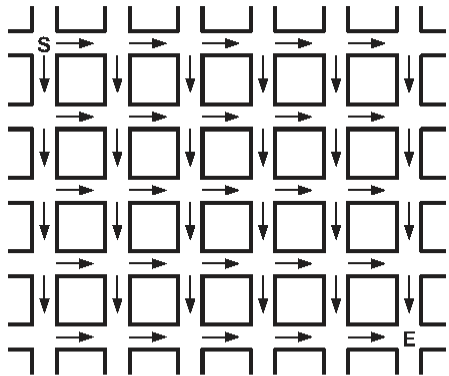


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

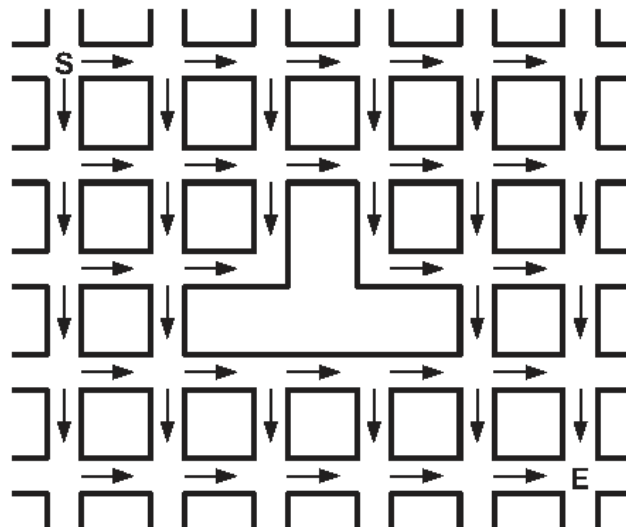


Figure B.



7F90D8EB-31CF-407E-9BF9-09CAF9018A0A

SPRING2014_CSCI570_EXAM2

#134 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

5BADBA5B-352E-4A0A-B635-84EA8FD44461

SPRING2014_CSCI570_EXAM2

#134

9 of 10



Additional Space



C2358E95-602B-478D-A9D8-455CB00A39A4

SPRING2014_CSCI570_EXAM2

#134 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



50E277AE-C959-4A2C-AD4D-BC77CF6486C8

SPRING2014_CSCI570_EXAM2

#135 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



C2B80F5A-C62E-4702-B76A-427AE804BC61

SPRING2014_CSCI570_EXAM2

#135

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

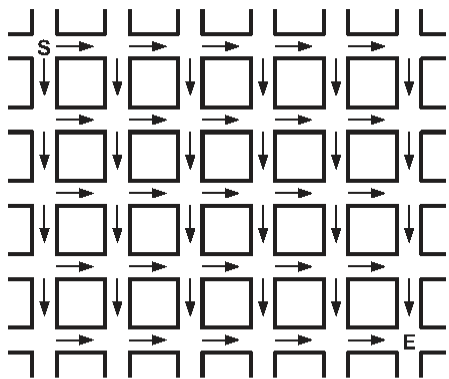


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



06D1002B-4290-400E-926D-108B94F36DB4

SPRING2014_CSCI570_EXAM2

#135 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

7FE1C3FC-42D4-429A-BDFF-19E41F530321

SPRING2014_CSCI570_EXAM2

#135

9 of 10



Additional Space



14726A1E-F4C9-4C96-8C1A-62DF969BF143

SPRING2014_CSCI570_EXAM2

#135 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



CD4EC33B-1208-46D4-876B-D959E592E3E8

SPRING2014_CSCI570_EXAM2

#136

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



16BF28AA-9730-45B3-8299-060BF422F593

SPRING2014_CSCI570_EXAM2

#136 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

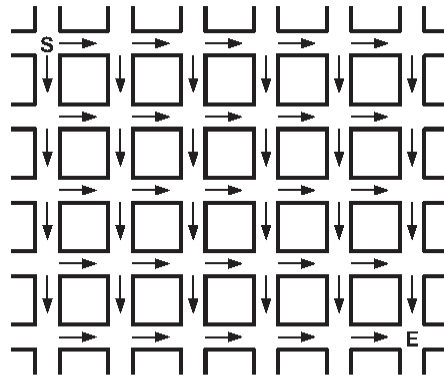


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

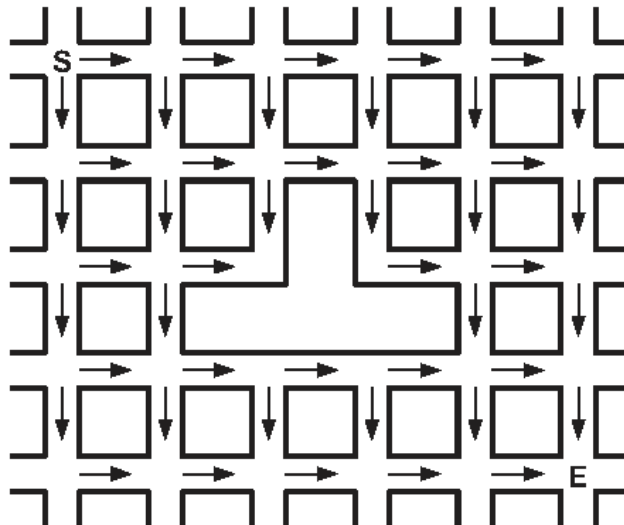


Figure B.



4BA4046A-EE26-4F92-911B-A05B267FB50E

SPRING2014_CSCI570_EXAM2

#136 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

B5450724-8419-4841-A4C9-F0D9135EB78B

SPRING2014_CSCI570_EXAM2

#136

9 of 10



Additional Space



CE86A5DC-7EB1-46D8-BCE6-0BAE6350163C

SPRING2014_CSCI570_EXAM2

#136 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



3380C82E-493A-440E-91CC-419F5E7A437D

SPRING2014_CSCI570_EXAM2

#137 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



C8C8415D-90B4-41A3-8119-C94F48923111

SPRING2014_CSCI570_EXAM2

#137 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

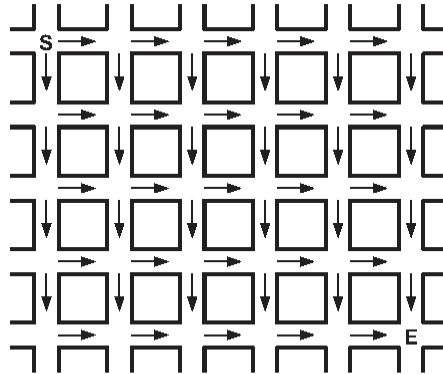


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

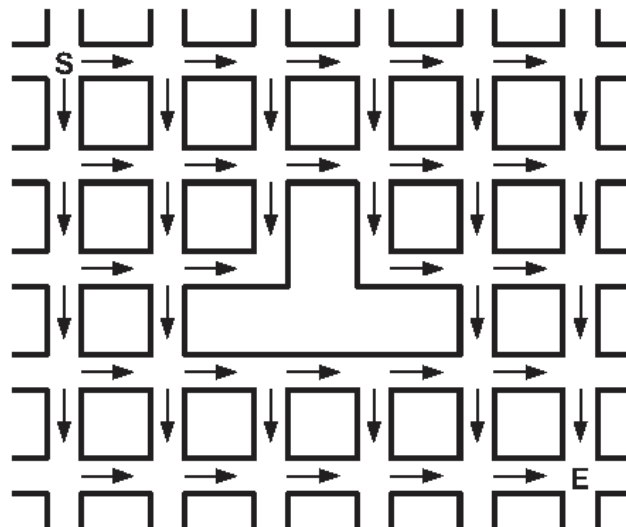


Figure B.



AED58202-341F-49B0-A933-FEE84014838F

SPRING2014_CSCI570_EXAM2

#137 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

3C36B0E9-A8C9-4693-B4D2-90F62B42CFDD

SPRING2014_CSCI570_EXAM2

#137

9 of 10



Additional Space



5642B6DC-C5FB-4ABF-9446-542CF6ACC802

SPRING2014_CSCI570_EXAM2

#137 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



783D8C36-0900-41FA-AE9B-A28286EE15E8

SPRING2014_CSCI570_EXAM2

#138

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



7DE96AC5-D413-4986-A0F3-4BE8E6D3BEEB

SPRING2014_CSCI570_EXAM2

#138 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

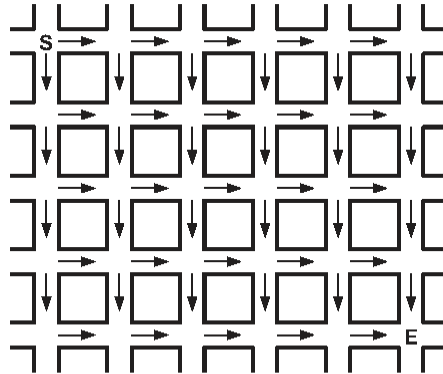


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

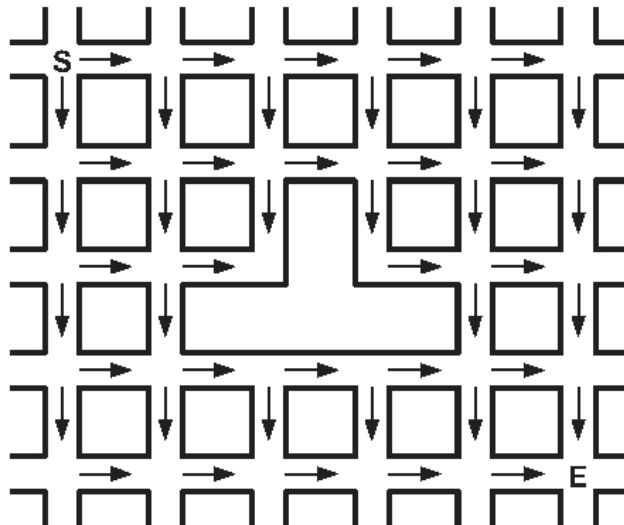


Figure B.



D6053EBD-1A92-4C0A-B217-40506E3A67A7

SPRING2014_CSCI570_EXAM2

#138 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9B6C5D80-572B-4C35-904C-C9FB36012869

SPRING2014_CSCI570_EXAM2

#138

9 of 10



Additional Space



541A5A91-F1F8-4A08-93CE-AFB48199C2EE

SPRING2014_CSCI570_EXAM2

#138 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



3A58EC73-0C48-4E00-A613-13579CFFDE91

SPRING2014_CSCI570_EXAM2

#139 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



B51475B8-34B4-4EE5-AC2B-29FA3905F11F

SPRING2014_CSCI570_EXAM2

#139 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

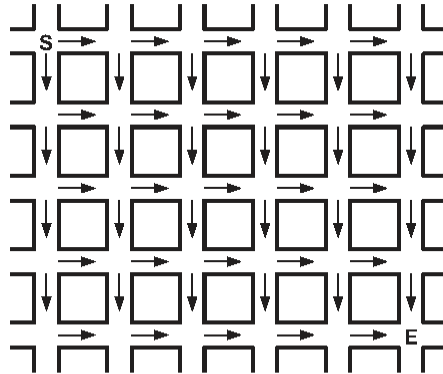


Figure A.



12BAC1EB-6940-49D0-9249-9191D2C5D484

SPRING2014_CSCI570_EXAM2

#139 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

60EAA54C-4639-47C4-8BB7-FC98BC65E12D

SPRING2014_CSCI570_EXAM2

#139

9 of 10



Additional Space



6A6B5832-B868-4347-BE46-C8E9068BEEB2

SPRING2014_CSCI570_EXAM2

#139 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



84DD6A9F-4B6E-40D7-AA81-2450A5CD5E4B

SPRING2014_CSCI570_EXAM2

#140 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



FE89977E-8BB8-402D-87BB-0C32C9CC1826

SPRING2014_CSCI570_EXAM2

#140 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



7FC23289-8FE3-42EE-9D19-F946C0DD9261

SPRING2014_CSCI570_EXAM2

#140 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9CC52F9B-9536-462E-9960-B62951414891

SPRING2014_CSCI570_EXAM2

#140

9 of 10



Additional Space



3FC8A8B6-2F9D-4E04-84D9-8FEA0CC107D8

SPRING2014_CSCI570_EXAM2

#140 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



90AAD177-56DA-47A5-A412-9BB70A21F0A5

SPRING2014_CSCI570_EXAM2

#141 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



294E3C99-9A0C-411C-8CCA-5A6A73FD6470

SPRING2014_CSCI570_EXAM2

#141 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

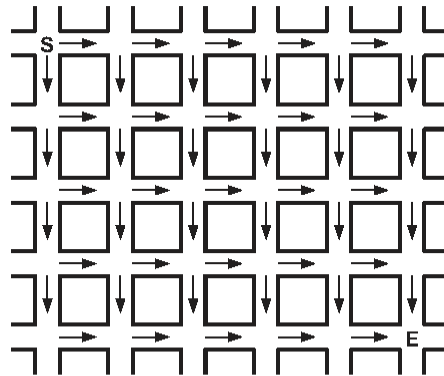


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

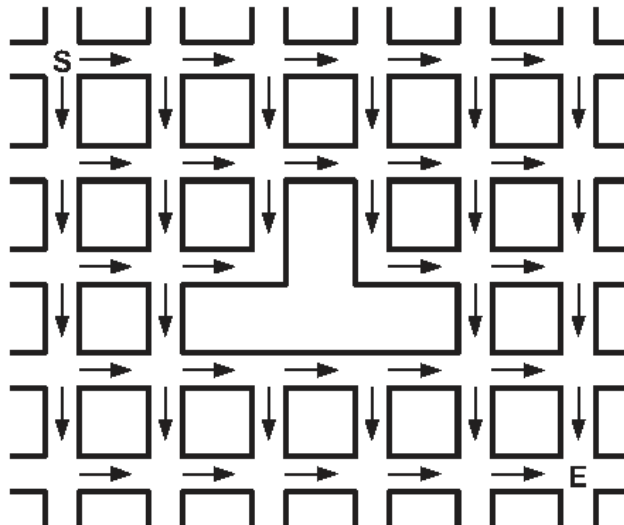


Figure B.



68F80C7C-255C-456B-93BB-5F34C9FBDDF8

SPRING2014_CSCI570_EXAM2

#141 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

2029B34B-9B67-4346-82AF-7A545BC3E23A

SPRING2014_CSCI570_EXAM2

#141

9 of 10



Additional Space



B742209D-C96C-4772-A558-38F3A4DCD643

SPRING2014_CSCI570_EXAM2

#141 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



526154FF-BF65-40F3-AD41-42359301C423

SPRING2014_CSCI570_EXAM2

#142 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



9FADE690-61A3-4186-B3C3-D64FBCCA7A23

SPRING2014_CSCI570_EXAM2

#142 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

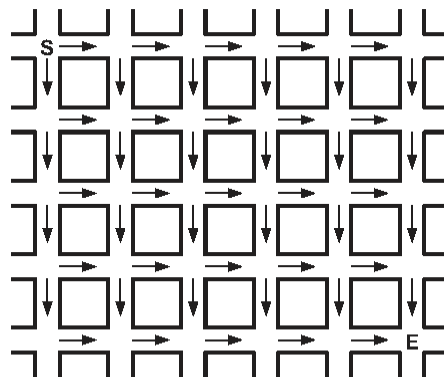


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



9EE922D3-C28B-4163-9A12-D2FD892E85DB

SPRING2014_CSCI570_EXAM2

#142 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

95E24803-3E37-44D5-850F-D679BE9241F8

SPRING2014_CSCI570_EXAM2

#142

9 of 10



Additional Space



67294EC8-742A-4ECD-A574-579FD23920C8

SPRING2014_CSCI570_EXAM2

#142 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



CA742F5F-5771-4975-9AAA-4463AC2A7216

SPRING2014_CSCI570_EXAM2

#143

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



2D13B9EF-F4D1-4029-97CD-9BAFE290D971

SPRING2014_CSCI570_EXAM2

#143 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

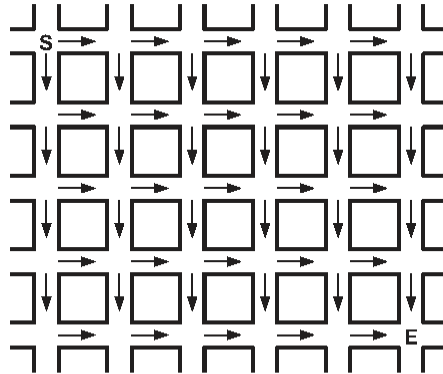


Figure A.



B90487B6-4E89-4549-9996-4C468B0A2F19

SPRING2014_CSCI570_EXAM2

#143 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

6664297B-EABE-4CFE-8677-DEF0D78BA6E1

SPRING2014_CSCI570_EXAM2

#143

9 of 10



Additional Space



6B405887-FE5F-4522-8881-2890FAF00109

SPRING2014_CSCI570_EXAM2

#143 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



A1446FF1-44DC-49FF-945F-E2230AE46C02

SPRING2014_CSCI570_EXAM2

#144 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

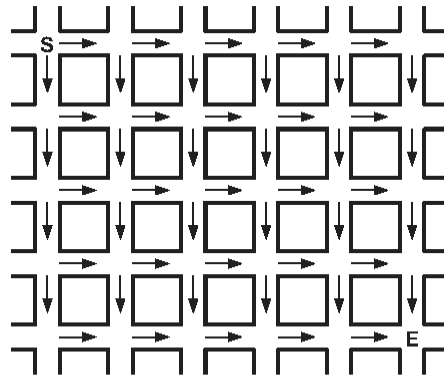


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

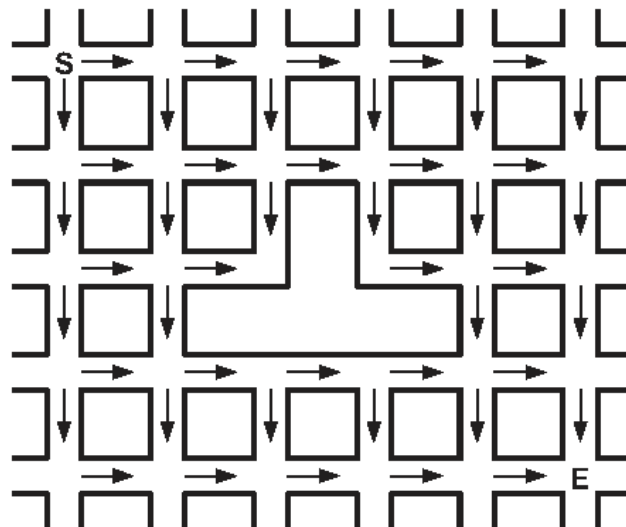


Figure B.



EE97E861-88E7-4345-966D-2EEFDE66B456

SPRING2014_CSCI570_EXAM2

#144 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

1C56289D-A5C2-40D3-8100-2CE603AA2F9F

SPRING2014_CSCI570_EXAM2

#144

9 of 10



Additional Space



3948F837-BB6C-47CB-8AC7-25838FBA4D6E

SPRING2014_CSCI570_EXAM2

#144 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



2BA1B433-55F3-42C7-8D4A-368A58B4DE71

SPRING2014_CSCI570_EXAM2

#145

2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



5323DE56-6B88-4113-A148-F33D08C8ABFF

SPRING2014_CSCI570_EXAM2

#145

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



08D932F5-AF24-493A-A993-8A8D3D1F3099

SPRING2014_CSCI570_EXAM2

#145

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

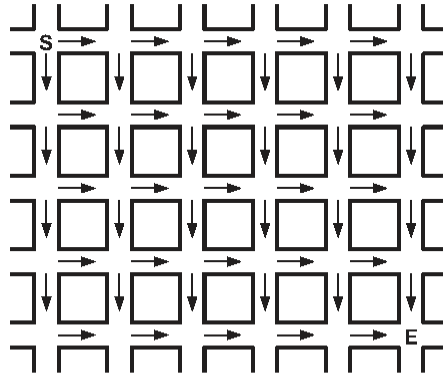


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

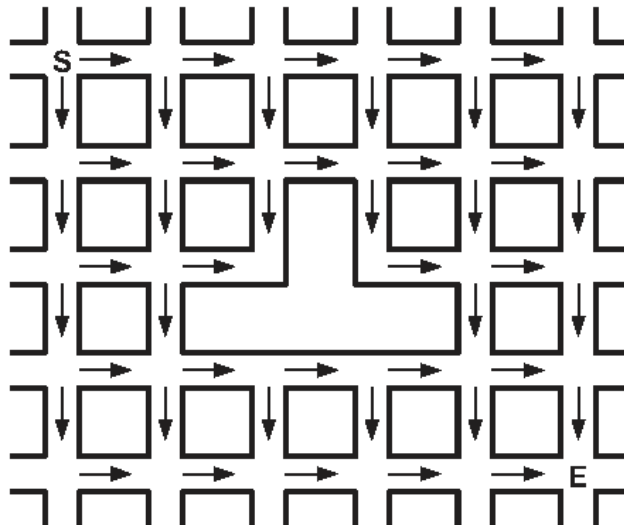


Figure B.



8026A9F7-A285-45F4-97D6-A4129DEBC9BE

SPRING2014_CSCI570_EXAM2

#145 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9C079481-34A0-4977-BD78-F92CFA325E7A

SPRING2014_CSCI570_EXAM2

#145

9 of 10



Additional Space



0CB72015-4ED3-4368-B584-9ED82D4B5582

SPRING2014_CSCI570_EXAM2

#145 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



75F9B976-0AED-40CC-9CB7-26313C3213FF

SPRING2014_CSCI570_EXAM2

#146

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

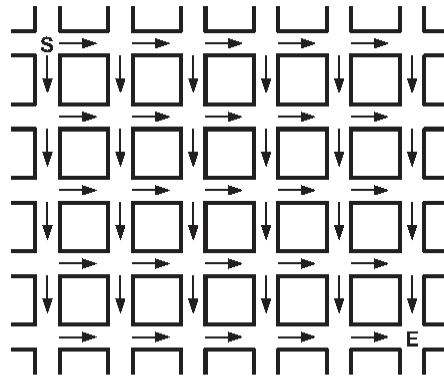


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

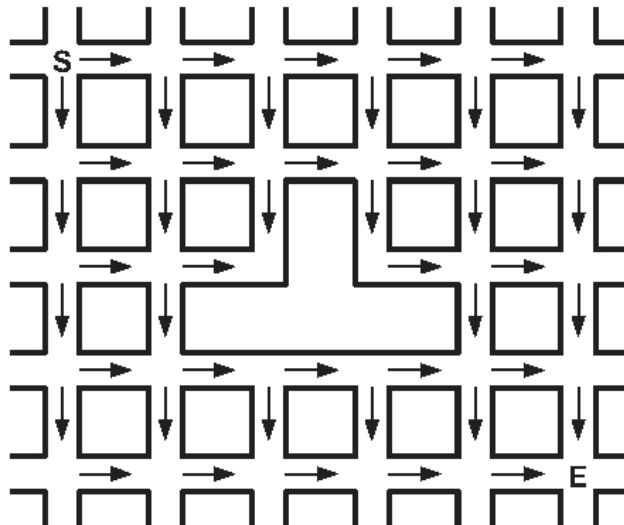


Figure B.



3795806F-964A-4E4B-AC44-44441F90B6D7

SPRING2014_CSCI570_EXAM2

#146 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

A8BE0299-BCBC-460C-A599-90B70F4E08A3

SPRING2014_CSCI570_EXAM2

#146

9 of 10



Additional Space



993BC879-322F-451A-BB80-9EE9A655F7D3

SPRING2014_CSCI570_EXAM2

#146 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



16678DA5-0B2B-4DCC-B45B-FC32E4727843

SPRING2014_CSCI570_EXAM2

#147 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

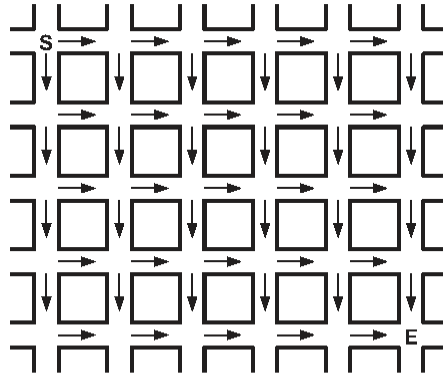


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

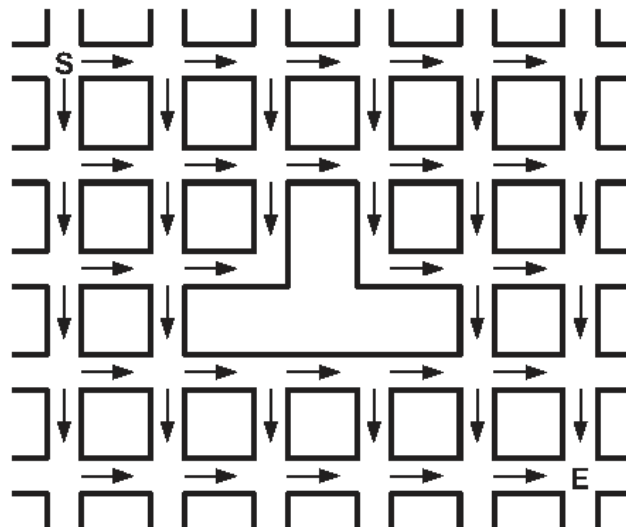


Figure B.



02EA175F-6416-456F-A4EA-6B729C667B1B

SPRING2014_CSCI570_EXAM2

#147 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

4466937A-320B-40F2-87EA-F5CD1883B4B2

SPRING2014_CSCI570_EXAM2

#147

9 of 10



Additional Space



006C7F37-FE37-4BBF-BBBD-440571099082

SPRING2014_CSCI570_EXAM2

#147 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



C18BF2C7-0C49-42C5-8316-E5E78E3679C4

SPRING2014_CSCI570_EXAM2

#148 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



A380C1B2-8AE4-4ADA-A176-194E301F3A8F

SPRING2014_CSCI570_EXAM2

#148

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



A8834B48-9115-485A-B478-ADD7265DCCC0

SPRING2014_CSCI570_EXAM2

#148 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

644685B7-FF18-49B3-8D13-AB8227EE72EC

SPRING2014_CSCI570_EXAM2

#148

9 of 10



Additional Space



6465EC6D-935F-4C26-AFF7-D9268B62FC6B

SPRING2014_CSCI570_EXAM2

#148 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



E25C2DED-1B57-464E-BBD7-1CE799EE7112

SPRING2014_CSCI570_EXAM2

#149 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



1F15BBB0-ACA6-4F2F-8BC2-8DE806DD4EF9

SPRING2014_CSCI570_EXAM2

#149

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

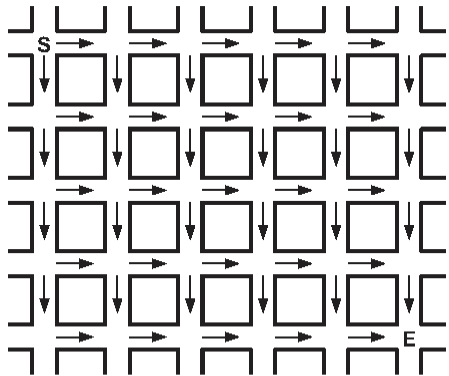


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

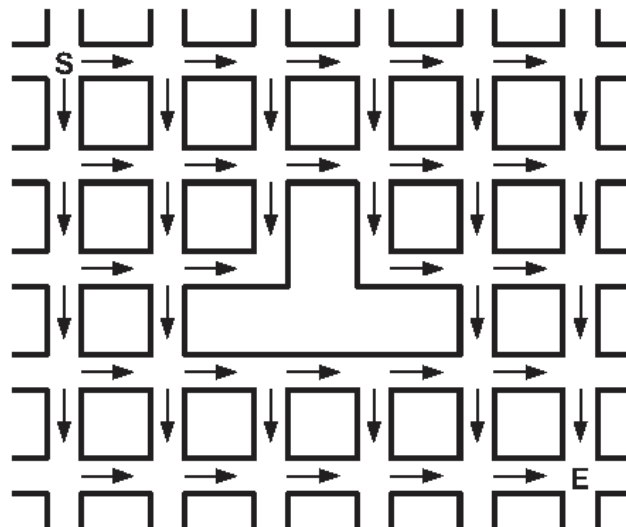


Figure B.



259C27B8-6076-4E3B-9B92-1E300F6C7C2D

SPRING2014_CSCI570_EXAM2

#149 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

6F98C8FB-AF31-4F61-A07E-2247674A9084

SPRING2014_CSCI570_EXAM2

#149

9 of 10



Additional Space



9382390C-EBE5-467D-8DF2-179CFD706700

SPRING2014_CSCI570_EXAM2

#149 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



89E99756-2F58-44FA-93DB-4618B12500D7

SPRING2014_CSCI570_EXAM2

#150 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



B87BE115-EB40-43FC-8154-9D0BB3417940

SPRING2014_CSCI570_EXAM2

#150 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



0EC458D6-A308-4B8C-BBBB-9BDAEC5A5D8C

SPRING2014_CSCI570_EXAM2

#150 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

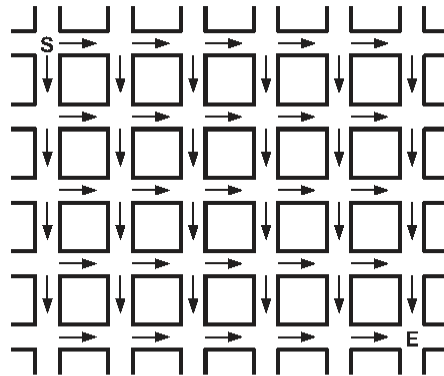


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

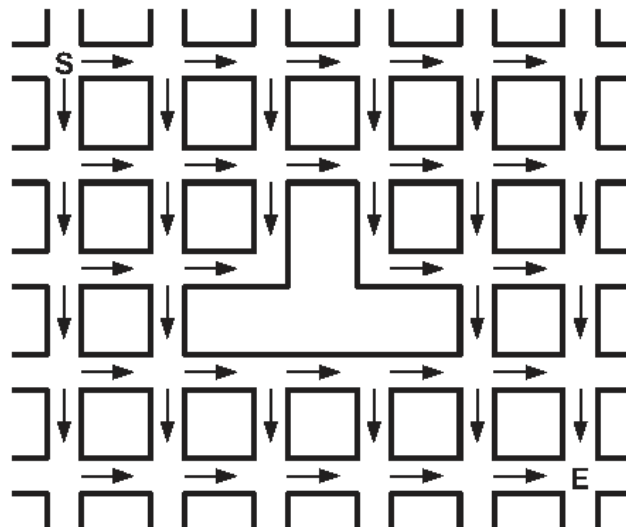


Figure B.



D2BF0E44-15EF-4E79-ACD1-DE9E1C237CA6

SPRING2014_CSCI570_EXAM2

#150 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

23F7B33C-9B8D-4621-908F-A9ADCE2D3B28

SPRING2014_CSCI570_EXAM2

#150

9 of 10



Additional Space



0C8D3BA8-6F84-44B6-B414-99BC30F3BB3E

SPRING2014_CSCI570_EXAM2

#150 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



C0A06E02-FF00-4ACD-953E-6F83A6797D6A

SPRING2014_CSCI570_EXAM2

#151 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



1033FB3A-75F4-4D1F-9E52-4350E7F8F543

SPRING2014_CSCI570_EXAM2

#151 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

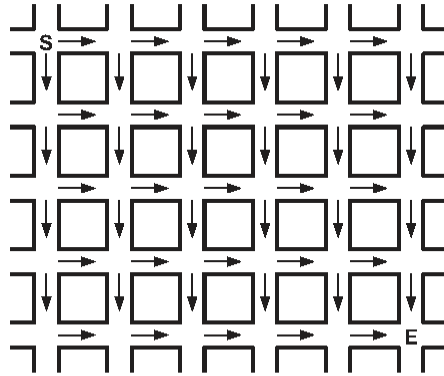


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



D0D58AEA-2A1E-49BA-AD14-D375F8926777

SPRING2014_CSCI570_EXAM2

#151 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9E53507E-9A36-4C99-BDDE-AD805826DF70

SPRING2014_CSCI570_EXAM2

#151

9 of 10



Additional Space



8599C035-D633-446F-A6EB-2E0D10123CF0

SPRING2014_CSCI570_EXAM2

#151 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



C541D3F8-D029-489E-964D-344D78F9C78E

SPRING2014_CSCI570_EXAM2

#152 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\cdot, \cdot)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\cdot, \cdot)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



05D858E5-2BA5-4C65-BFD0-0D47442A017F

SPRING2014_CSCI570_EXAM2

#152 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

B9448704-E868-403D-A88C-E7799C8AF6DC

SPRING2014_CSCI570_EXAM2

#152

9 of 10



Additional Space



AF2C260B-958F-40B2-BA97-E411256DCB99

SPRING2014_CSCI570_EXAM2

#152 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



D2DAA266-719B-4A31-B01F-5F0601DB6962

SPRING2014_CSCI570_EXAM2

#153 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



D3F16DF4-0824-4CF0-905D-E9D6AE3942D6

SPRING2014_CSCI570_EXAM2

#153 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

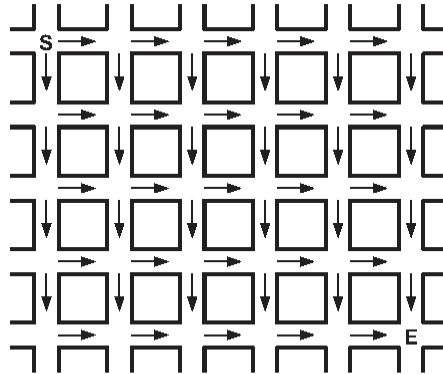


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

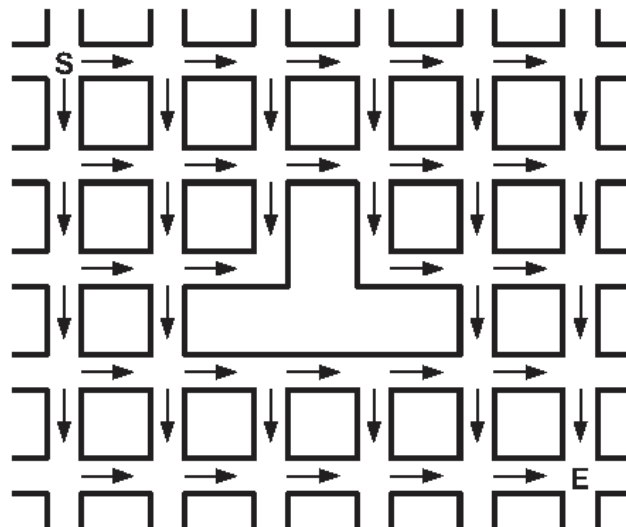


Figure B.



C7F7C3F1-A10F-4F45-B7B6-4854E6DCB8FE

SPRING2014_CSCI570_EXAM2

#153 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

AAEF0810-8605-4195-B4AC-EBE1F4E64B5C

SPRING2014_CSCI570_EXAM2

#153

9 of 10



Additional Space



8C134F70-C18D-460E-8FE1-3E68B9237F29

SPRING2014_CSCI570_EXAM2

#153 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



0C6115C9-8A2D-4E82-A864-536571EB6B89

SPRING2014_CSCI570_EXAM2

#154 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



A3DD11EE-2A27-4F99-A824-612DC92BF609

SPRING2014_CSCI570_EXAM2

#154 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

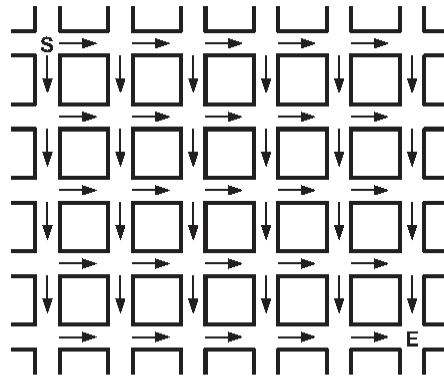


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



D922B114-5807-427B-A5E0-501747A08B24

SPRING2014_CSCI570_EXAM2

#154 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

56C440F2-AAA6-46F4-A5BA-0F7CA7DBC9A4

SPRING2014_CSCI570_EXAM2

#154 9 of 10



Additional Space



ADE1307E-C4BA-4D59-AB9D-622FF9F3232A

SPRING2014_CSCI570_EXAM2

#154 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



E65F909E-458A-4E95-B68B-3108ED48001D

SPRING2014_CSCI570_EXAM2

#155 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



AF5CA255-D8DB-4440-8622-39A203D4844B

SPRING2014_CSCI570_EXAM2

#155 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



925141AD-6605-4283-867D-E834D2A82065

SPRING2014_CSCI570_EXAM2

#155 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

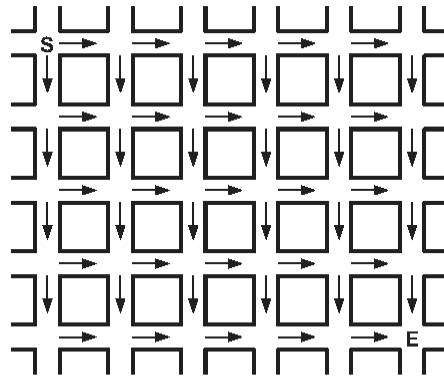


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

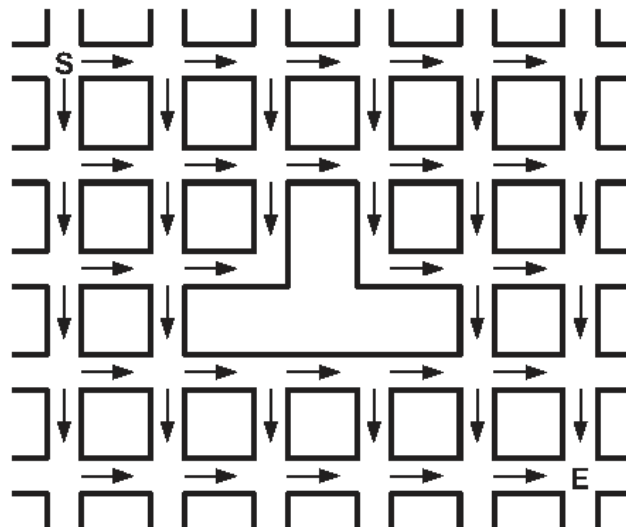


Figure B.



2D51E26E-6F8C-41D3-8218-0226F891B05C

SPRING2014_CSCI570_EXAM2

#155 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

B8E5C9F3-B8B0-41C7-8EA3-7ABF78368CF4

SPRING2014_CSCI570_EXAM2

#155

9 of 10



Additional Space



68343DDE-9ED8-4EB8-B75B-E47068A48243

SPRING2014_CSCI570_EXAM2

#155 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



98A3FA06-D917-4195-9F3C-622B44D304BC

SPRING2014_CSCI570_EXAM2

#156

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

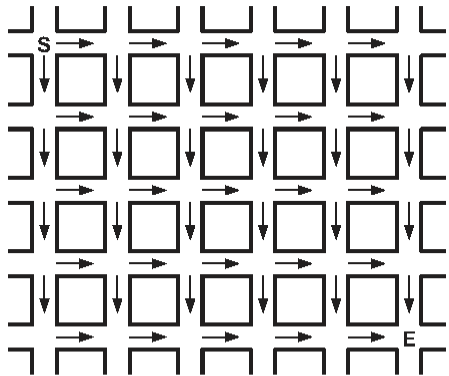


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

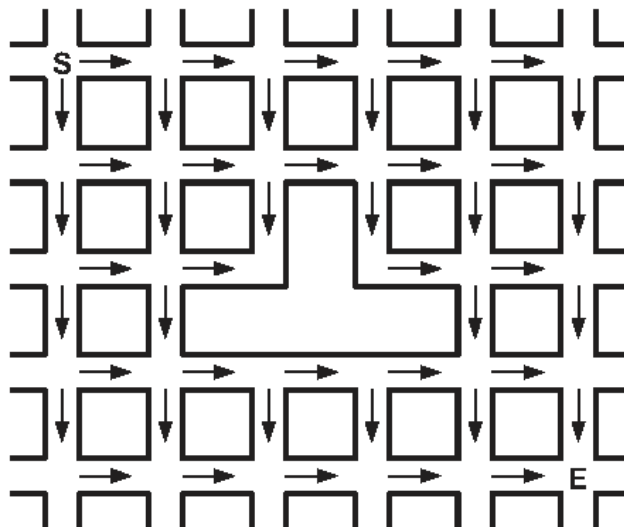


Figure B.



912C380E-FDEE-429F-AB3D-D0AB84CE5F7F

SPRING2014_CSCI570_EXAM2

#156 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

46DD87DD-74BE-4C2C-BC69-211B6C0FDC2E

SPRING2014_CSCI570_EXAM2

#156

9 of 10



Additional Space



FA3BB73D-F07F-4C85-88D8-78EBC151E685

SPRING2014_CSCI570_EXAM2

#156 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



5E9A34BD-302F-4526-AD32-C9EAD81B9080

SPRING2014_CSCI570_EXAM2

#157 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



C7A92597-0CA3-4C03-80B4-EFA77B82C077

SPRING2014_CSCI570_EXAM2

#157 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



46F19976-E6A1-40F6-9066-7C1BD3AE36BF

SPRING2014_CSCI570_EXAM2

#157 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

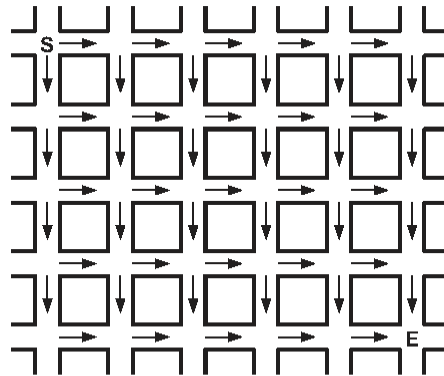


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

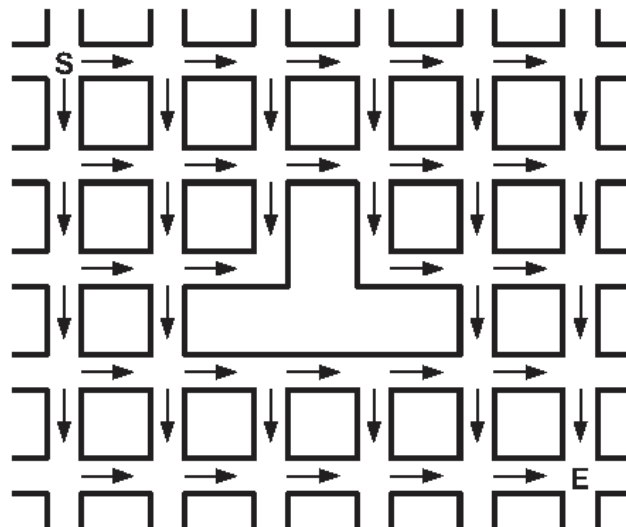


Figure B.



772E24C5-427A-47D5-A645-1DE4F0846A4B

SPRING2014_CSCI570_EXAM2

#157 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

50722F53-0DF3-4F7D-9B1E-385C7FD637AC

SPRING2014_CSCI570_EXAM2

#157

9 of 10



Additional Space



3F2075A2-D0B8-49FB-BB73-4F5F46976BB1

SPRING2014_CSCI570_EXAM2

#157 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



4393A3A9-54B3-4EF7-A0F3-CE768D11187C

SPRING2014_CSCI570_EXAM2

#158

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

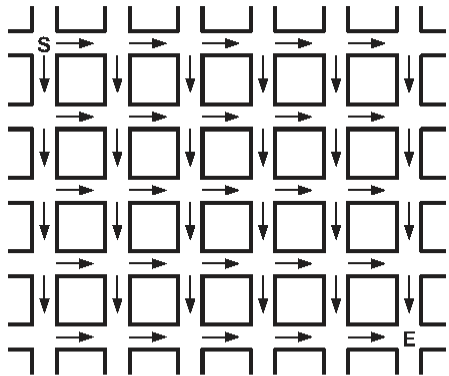


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



5825224F-D200-4197-AB60-6F94B64558D2

SPRING2014_CSCI570_EXAM2

#158 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

64E9B1B6-A4D4-4DD8-81E6-B92111E70BD6

SPRING2014_CSCI570_EXAM2

#158

9 of 10



Additional Space



78685AAA-3F0E-449B-9C8C-F47CBD22100E

SPRING2014_CSCI570_EXAM2

#158 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



EFF788D2-AEB4-4E08-A131-814256DE4BB2

SPRING2014_CSCI570_EXAM2

#159 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

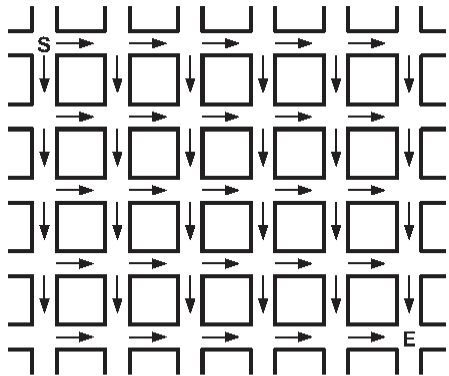


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



0B94313F-B652-4AD7-817F-CC5C4A6045D7

SPRING2014_CSCI570_EXAM2

#159 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

54E4191B-09BC-4ED5-8B16-23691C583437

SPRING2014_CSCI570_EXAM2

#159

9 of 10



Additional Space



8B85606E-8E3C-4A0E-B664-BCD299EFCD7

SPRING2014_CSCI570_EXAM2

#159 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



EEFBB619-22E1-407D-A8BA-0FEF6011C7ED

SPRING2014_CSCI570_EXAM2

#160 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



A772F67E-9AB9-4AF5-8C13-6D3C6CB102E8

SPRING2014_CSCI570_EXAM2

#160

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



D3E12319-E182-4A71-90A1-8FF57355267C

SPRING2014_CSCI570_EXAM2

#160 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

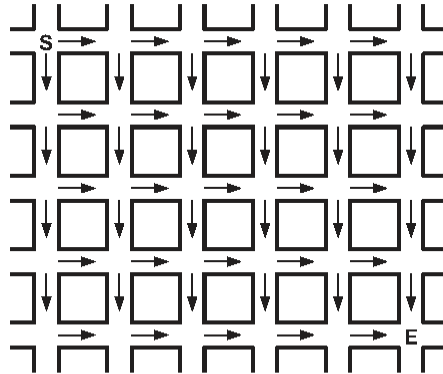


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



DA969D6C-D931-4D86-B257-B971F2905910

SPRING2014_CSCI570_EXAM2

#160 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

32A35AB5-410D-44B6-87D5-F07FE98379A7

SPRING2014_CSCI570_EXAM2

#160

9 of 10



Additional Space



FFF06EEB-A330-4D27-8C02-812D96340BFF

SPRING2014_CSCI570_EXAM2

#160 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



065CBCEA-84F3-4791-ADCA-51124F1BE6AC

SPRING2014_CSCI570_EXAM2

#161 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



0A5E31E7-86E7-474B-8259-4A272EFA1499

SPRING2014_CSCI570_EXAM2

#161

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

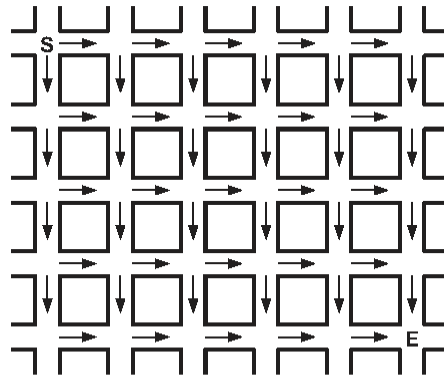


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



AF786D9F-09C2-402B-B775-4C711C0DCA10

SPRING2014_CSCI570_EXAM2

#161 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

04C6B0AE-5ACE-4302-A405-A276457CD6A2

SPRING2014_CSCI570_EXAM2

#161

9 of 10



Additional Space



721D7C19-AC52-43C7-985D-A159E8BFB46F

SPRING2014_CSCI570_EXAM2

#161 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



9E876CA5-FF86-4830-8AD0-77D203A562E7

SPRING2014_CSCI570_EXAM2

#162 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



DF573155-5816-4F79-9371-2F56B1BA6125

SPRING2014_CSCI570_EXAM2

#162 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

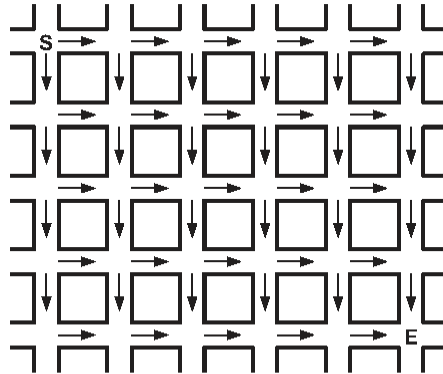


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



00950A56-631F-4DAA-B1F5-E91B09B6C029

SPRING2014_CSCI570_EXAM2

#162 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

426C9F2A-A223-4799-ACD2-5A8A454BA1B7

SPRING2014_CSCI570_EXAM2

#162

9 of 10



Additional Space



0445A8FA-6B0D-4921-B31D-7DC8FFAF864D

SPRING2014_CSCI570_EXAM2

#162 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



34A492F8-BD55-42D8-9DC5-DB47C8A654F5

SPRING2014_CSCI570_EXAM2

#163 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

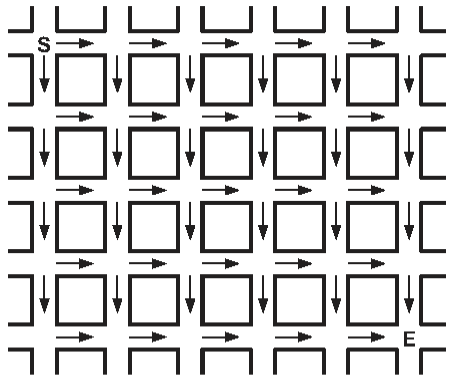


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



F21C664F-379A-403C-9D44-5D58CE3E29AB

SPRING2014_CSCI570_EXAM2

#163 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

F897BC02-F4B2-40C3-8094-94F30AE99D0B

SPRING2014_CSCI570_EXAM2

#163

9 of 10



Additional Space



F53F63CB-E713-4F0A-A152-FE571B25657D

SPRING2014_CSCI570_EXAM2

#163 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



0348F846-C372-4B19-8123-F7156995A890

SPRING2014_CSCI570_EXAM2

#164

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



1EF7B272-4B48-400B-9360-A67795158926

SPRING2014_CSCI570_EXAM2

#164

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

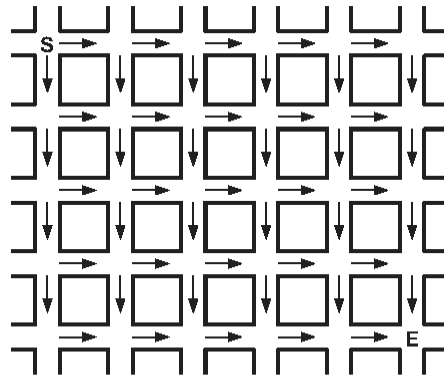


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



B6B18BAC-CE03-44B2-86A3-CA143CE744A5

SPRING2014_CSCI570_EXAM2

#164 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

7FA8D0A3-D300-4C04-B3F9-ABEF00511BB4

SPRING2014_CSCI570_EXAM2

#164

9 of 10



Additional Space



FAA6D0F8-3F69-4FB8-AE16-C6A78579D159

SPRING2014_CSCI570_EXAM2

#164 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



12CCB65C-C6F8-44C6-90F5-407CB95EF630

SPRING2014_CSCI570_EXAM2

#165 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



4A4B3E10-CBC8-4201-8ABE-4912DE7719E7

SPRING2014_CSCI570_EXAM2

#165 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



A615B2CF-7D49-410A-8C0D-72A92C93E47B

SPRING2014_CSCI570_EXAM2

#165 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

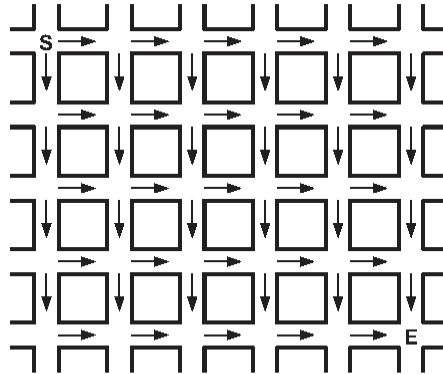


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



9178A2E9-B1A1-49EF-A2E5-CE3F209CA4F5

SPRING2014_CSCI570_EXAM2

#165 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

541BAADB-FF3F-4F31-A67F-2C8EF4A78551

SPRING2014_CSCI570_EXAM2

#165

9 of 10



Additional Space



86CE3ECE-59B0-4911-AD51-7A3BF681D77A

SPRING2014_CSCI570_EXAM2

#165 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



FE57974D-65D0-4142-8B14-CC802F0C4C31

SPRING2014_CSCI570_EXAM2

#166

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

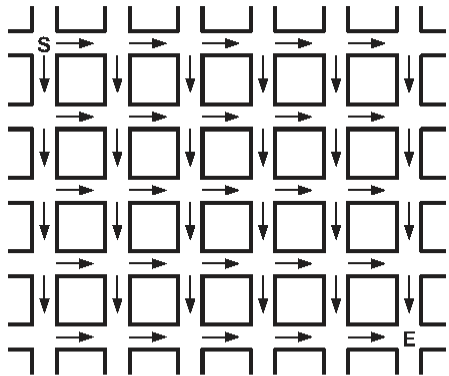


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



FB487F55-B481-48A2-8C11-2BDEFBD2DE14

SPRING2014_CSCI570_EXAM2

#166 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

2B470AB3-5A8F-44BE-A151-E4FF95077A12

SPRING2014_CSCI570_EXAM2

#166

9 of 10



Additional Space



9C43FD89-A09F-42B1-9037-403BC1FC8938

SPRING2014_CSCI570_EXAM2

#166 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



B5BF492B-CE08-4166-B5E3-C179241B6003

SPRING2014_CSCI570_EXAM2

#167 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



711B3198-51D0-4B66-9BB2-0A211448D89C

SPRING2014_CSCI570_EXAM2

#167 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



192CEA9B-591C-4DE0-A554-6D37B59BC9F1

SPRING2014_CSCI570_EXAM2

#167 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

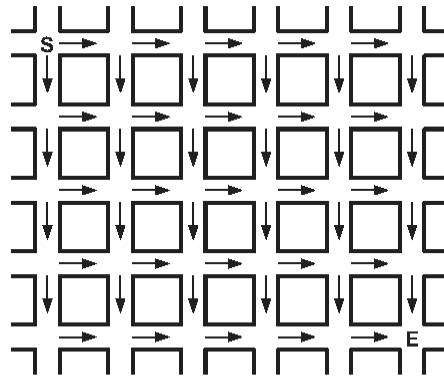


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

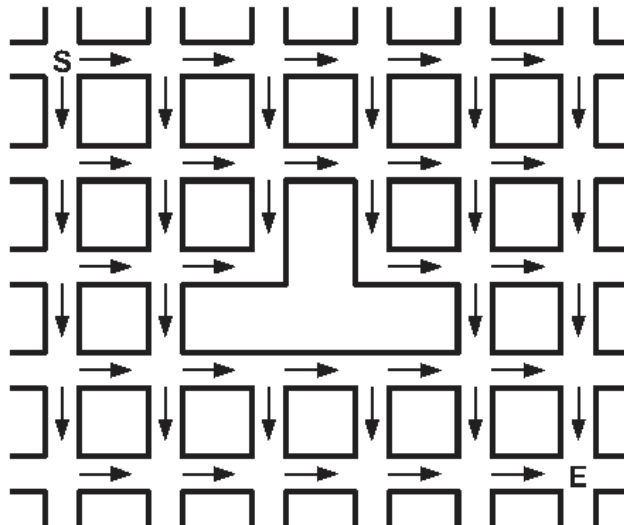


Figure B.



D863703E-9D1A-40B1-B51B-AA9226719252

SPRING2014_CSCI570_EXAM2

#167 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

A0CF29AD-CBC5-471D-A8F0-9159D6599155

SPRING2014_CSCI570_EXAM2

#167

9 of 10



Additional Space



CC4F9903-3797-4478-BBFE-AA3B9DE71421

SPRING2014_CSCI570_EXAM2

#167 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



7483E624-2C63-4D6A-B7D0-D8D7F83A50EB

SPRING2014_CSCI570_EXAM2

#168

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



49FA384B-A155-4F15-918C-E9FC518AE8EC

SPRING2014_CSCI570_EXAM2

#168

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

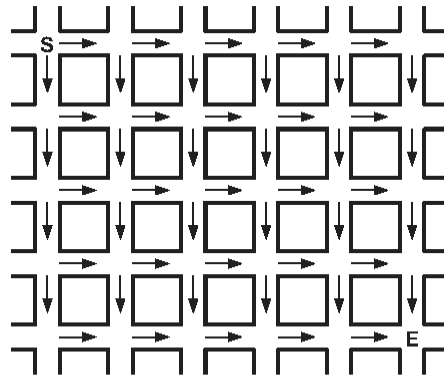


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



5C301D30-7204-49D0-BFA2-BC45D0C5A588

SPRING2014_CSCI570_EXAM2

#168 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

E3804F27-B769-48F6-BCE4-0B7ADC78940E

SPRING2014_CSCI570_EXAM2

#168

9 of 10



Additional Space



509396DE-F6AC-47B1-8D55-303691674C3E

SPRING2014_CSCI570_EXAM2

#168 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



E7C98D2A-B5FE-4A97-A77F-F6D476425402

SPRING2014_CSCI570_EXAM2

#169

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

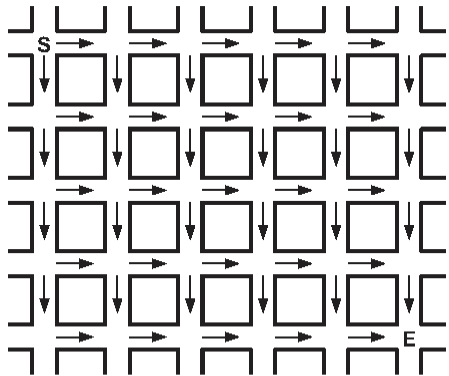


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

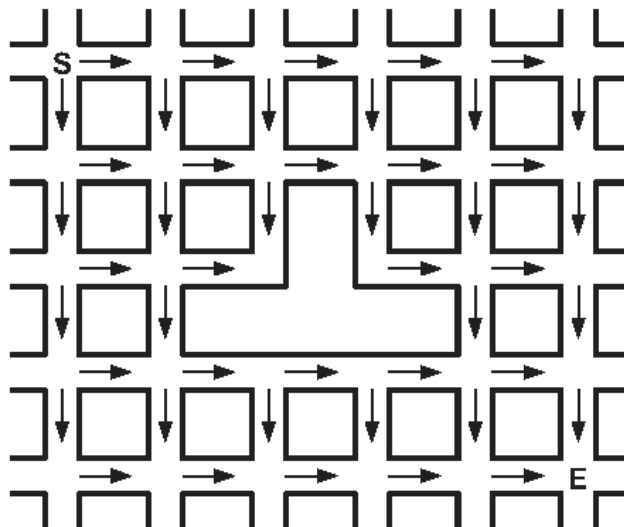


Figure B.



9A31F2B9-66E5-4348-BF24-7AC6DD1D48CA

SPRING2014_CSCI570_EXAM2

#169 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

2BEC6131-19D6-47C4-949D-50449DC54963

SPRING2014_CSCI570_EXAM2

#169

9 of 10



Additional Space



1FB16BAE-BCA1-4CE1-B4DE-6C490B44A24C

SPRING2014_CSCI570_EXAM2

#169 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



978EDB9C-F8B8-4ECA-AAD9-EA790E93C74F

SPRING2014_CSCI570_EXAM2

#170 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



66E4C689-15E2-416C-9C9F-0B157D89411C

SPRING2014_CSCI570_EXAM2

#170

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

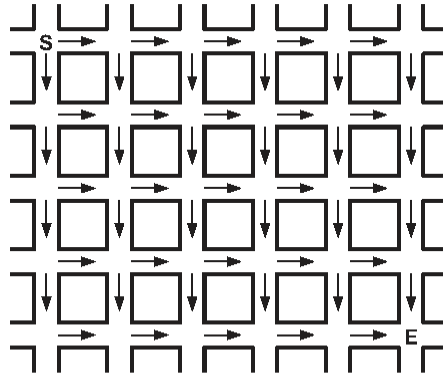


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



7133E3EB-7564-49BD-AE0E-1152AB19C289

SPRING2014_CSCI570_EXAM2

#170 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

D2F0C5B1-055B-499E-B9C7-7CFF15794788

SPRING2014_CSCI570_EXAM2

#170

9 of 10



Additional Space



C0EF9D3B-2843-40E3-A7E5-826C771DE998

SPRING2014_CSCI570_EXAM2

#170 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



7F8A9F72-E4B1-43A0-99AB-EE9197CB473F

SPRING2014_CSCI570_EXAM2

#171 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



9EECB3D3-B155-4255-9AA5-B3C9512DA907

SPRING2014_CSCI570_EXAM2

#171 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

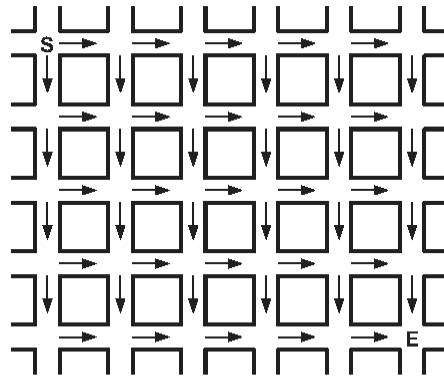


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



557270AE-C599-44C6-A6D0-87AFC54DB9E9

SPRING2014_CSCI570_EXAM2

#171 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

6810D4DB-0F76-4706-80FB-F1026410F6E3

SPRING2014_CSCI570_EXAM2

#171

9 of 10



Additional Space



9BBD0231-C646-4491-8914-274C29C08955

SPRING2014_CSCI570_EXAM2

#171 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



1E635A51-A99E-4C92-873C-9583708944A4

SPRING2014_CSCI570_EXAM2

#172 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

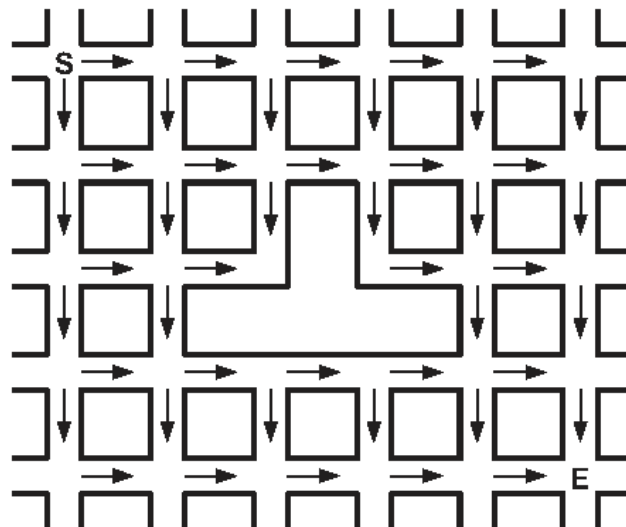


Figure B.



1E48E734-232A-4047-8202-5AF9E255E6B5

SPRING2014_CSCI570_EXAM2

#172 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

1D02A35A-92D7-4F3D-949B-0824E383EBBE

SPRING2014_CSCI570_EXAM2

#172

9 of 10



Additional Space



6452BB59-E8E6-4C90-8C23-40CD23FFF8B3

SPRING2014_CSCI570_EXAM2

#172 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



A692AFA0-00AA-40EE-B984-D71151D54555

SPRING2014_CSCI570_EXAM2

#173 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



B6FF2B39-FBBA-47A0-B9CA-0715BDB75D1D

SPRING2014_CSCI570_EXAM2

#173 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

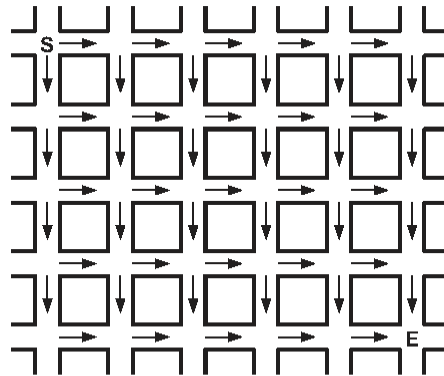


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

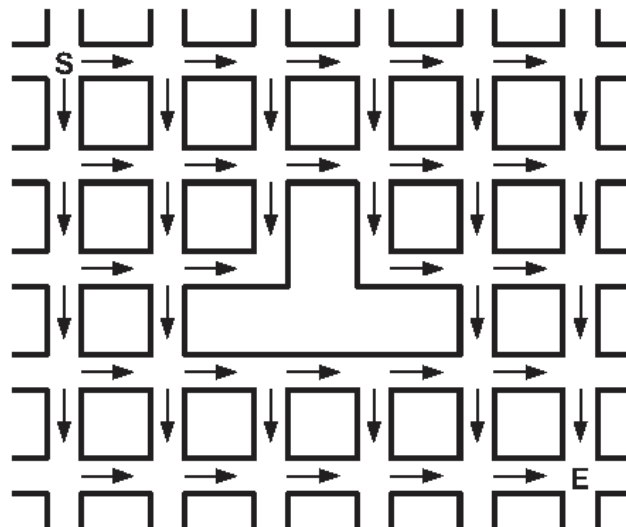


Figure B.



F9ACAF97-17D5-439E-B1E0-0177BCA58FCD

SPRING2014_CSCI570_EXAM2

#173 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

3CB375CA-0AD2-4BC8-BD9C-728156F979C3

SPRING2014_CSCI570_EXAM2

#173

9 of 10



Additional Space



0D21AA86-E9F0-4395-B004-A37F644BBD42

SPRING2014_CSCI570_EXAM2

#173 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



7AC0EC0C-86CC-4121-8F91-9FB61223C56E

SPRING2014_CSCI570_EXAM2

#174 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

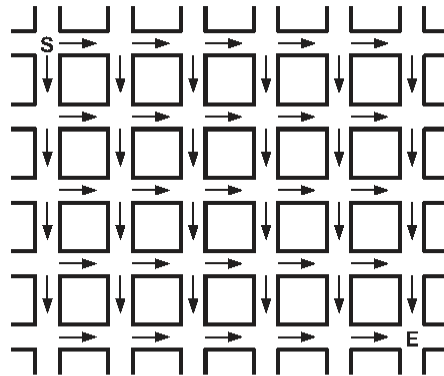


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



363A7A42-4C9E-4904-8D66-7C9A06ED4C2E

SPRING2014_CSCI570_EXAM2

#174 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

259FA1B6-49EB-4585-B7B1-6A86E1726D9F

SPRING2014_CSCI570_EXAM2

#174

9 of 10



Additional Space



8CE97D50-7963-4DBD-8232-CDFB3472A02E

SPRING2014_CSCI570_EXAM2

#174 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



B3AB56B5-3B29-4779-A1FE-C93475FFFE0

SPRING2014_CSCI570_EXAM2

#175 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

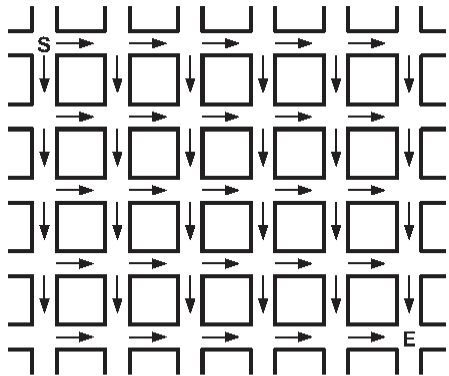


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



F732B358-B288-42C5-822F-48CFD69B4F6F

SPRING2014_CSCI570_EXAM2

#175 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

D0F146FE-6CF0-48F0-B822-B2A641B9BCC4

SPRING2014_CSCI570_EXAM2

#175

9 of 10



Additional Space



4056FE46-720F-4731-B047-DC3C10C23C40

SPRING2014_CSCI570_EXAM2

#175 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



2C100950-6429-41BC-A679-2CA7D4CC007D

SPRING2014_CSCI570_EXAM2

#176

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



AA4EF1C1-FDA7-4A9B-AD31-1E4CEFB48F31

SPRING2014_CSCI570_EXAM2

#176

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

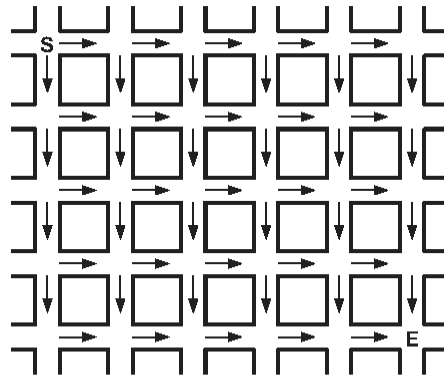


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



6C20F6B1-E122-4666-81F7-49526ABBFB0F

SPRING2014_CSCI570_EXAM2

#176 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

4063D6CF-40D3-483E-A636-6F932F87F097

SPRING2014_CSCI570_EXAM2

#176

9 of 10



Additional Space



185D79FF-6B91-489E-88EC-CB0B366CA71F

SPRING2014_CSCI570_EXAM2

#176 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



7883A173-DF53-4584-A941-E440D5F816A9

SPRING2014_CSCI570_EXAM2

#177 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



825C1675-083F-4F77-9D74-907BEADB8A64

SPRING2014_CSCI570_EXAM2

#177 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

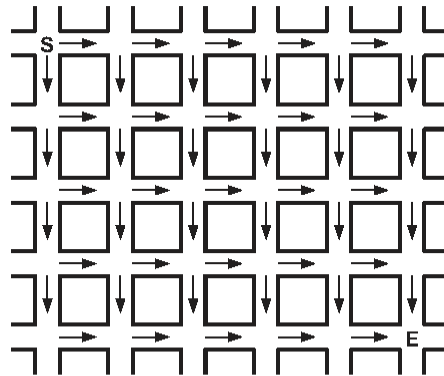


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

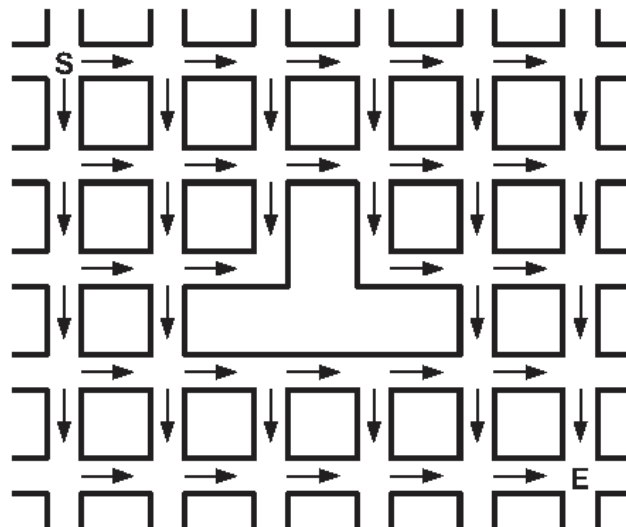


Figure B.



D8094294-0BED-4063-83CD-8163BDD7DF4D

SPRING2014_CSCI570_EXAM2

#177 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

FF2C5608-757B-4DE4-915F-27C9A62B2334

SPRING2014_CSCI570_EXAM2

#177

9 of 10



Additional Space



C0A16287-9AAA-4527-B296-46F413263F43

SPRING2014_CSCI570_EXAM2

#177 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



0F5571D6-F8DB-4F27-B473-7E86449427E4

SPRING2014_CSCI570_EXAM2

#178 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



6546D136-C07A-4952-A208-1382EC3E389E

SPRING2014_CSCI570_EXAM2

#178

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



2D3AF533-6920-42C1-9152-D6E62B1FAB8C

SPRING2014_CSCI570_EXAM2

#178

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

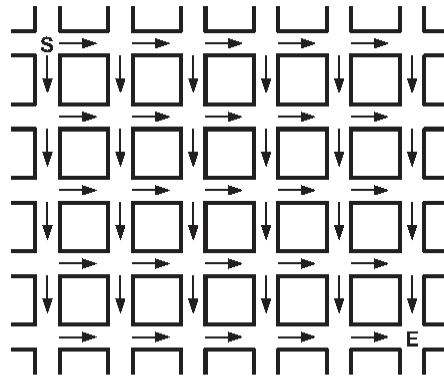


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



2816A3D4-89AF-43B0-A7EF-B8FE4BEAE858

SPRING2014_CSCI570_EXAM2

#178 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

E2662E6D-E455-411C-8374-609A1EABA124

SPRING2014_CSCI570_EXAM2

#178

9 of 10



Additional Space



574FB9AD-0871-406A-AF83-01B1DDDB447D

SPRING2014_CSCI570_EXAM2

#178 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



5AE4352E-58FA-441F-B5E2-A5E2D0C6DBE9

SPRING2014_CSCI570_EXAM2

#179 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



45E4924E-F12A-40BC-881D-05F41F492091

SPRING2014_CSCI570_EXAM2

#179

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



5FBBE8A1-3292-4CE4-9F73-EBECF08CD992

SPRING2014_CSCI570_EXAM2

#179

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

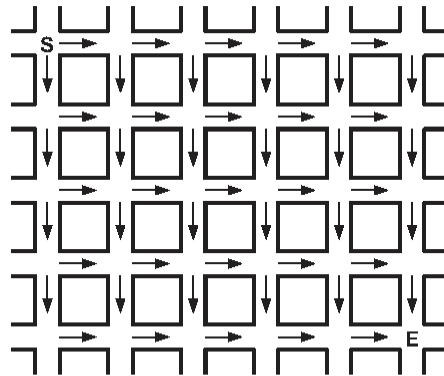


Figure A.

-

Figure B.



08645FB1-4EA6-4D72-A08E-CCBD6B3E66E1

SPRING2014_CSCI570_EXAM2

#179 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

3E8B8A0B-03BC-4157-953D-AB0A6E246B33

SPRING2014_CSCI570_EXAM2

#179

9 of 10



Additional Space



8C755807-E9FC-46A9-B866-27FD78C21FD7

SPRING2014_CSCI570_EXAM2

#179 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



6DC37E17-87E1-4F77-9B4D-78CE09617DA5

SPRING2014_CSCI570_EXAM2

#180 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



04CDB49A-7DF4-4500-A594-D7027CFFA003

SPRING2014_CSCI570_EXAM2

#180 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



4E8FBDC2-7181-4B1F-A01E-22E2CAC0EF97

SPRING2014_CSCI570_EXAM2

#180

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

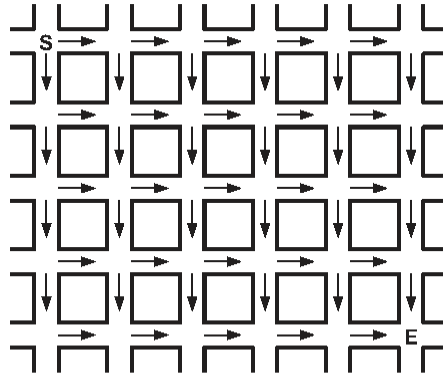


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

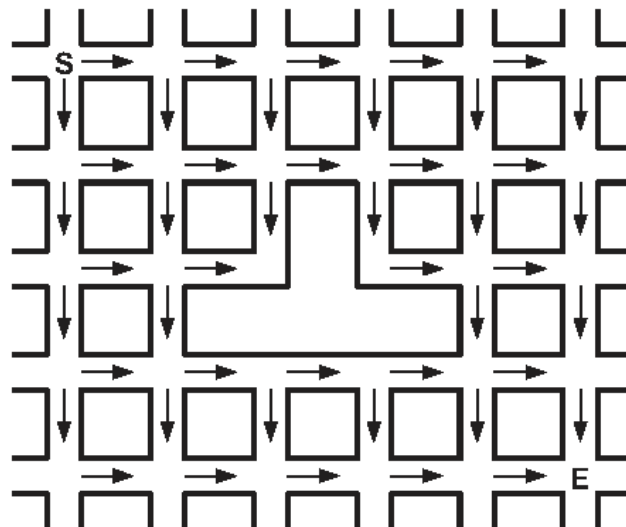


Figure B.



5D58D4B3-C07D-45C5-9F08-8D2D0B1F2DA2

SPRING2014_CSCI570_EXAM2

#180 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

CA7B52F8-B228-4FB0-9F3A-C4ABF57C5A6A

SPRING2014_CSCI570_EXAM2

#180

9 of 10



Additional Space



BAF187BE-3AF7-4684-8BFE-740046EBAB40

SPRING2014_CSCI570_EXAM2

#180 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



4A72945B-6F4F-4053-B23F-DBB7E9B32BA6

SPRING2014_CSCI570_EXAM2

#181 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

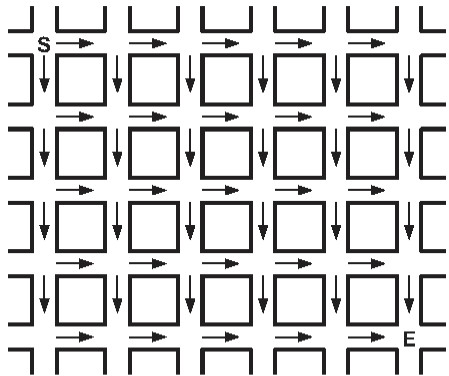


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



5F8128F6-F686-49F8-827D-B92C033FBBD7

SPRING2014_CSCI570_EXAM2

#181 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

5B960D77-0B45-4175-8353-E5C8621F9BBC

SPRING2014_CSCI570_EXAM2

#181

9 of 10



Additional Space



19C01FE6-FF74-4EBF-B749-D87A01B1A4E0

SPRING2014_CSCI570_EXAM2

#181 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



89A4546E-6A06-45CC-B666-E0DE47D3A4D8

SPRING2014_CSCI570_EXAM2

#182 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

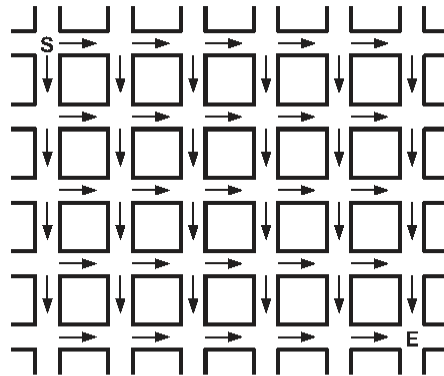


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



CE658413-222C-434A-AA03-6ABAB1B1C732

SPRING2014_CSCI570_EXAM2

#182 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

23C62529-D2C3-43AE-8C3A-970957F5F781

SPRING2014_CSCI570_EXAM2

#182

9 of 10



Additional Space



8AEF773E-D1F2-47A6-8A6F-919AC610BA45

SPRING2014_CSCI570_EXAM2

#182 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



C8C3A38E-D13C-41F0-AB64-2D9E5BCE3230

SPRING2014_CSCI570_EXAM2

#183 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



47B80098-F1D6-43E2-90B1-1F16237C55E3

SPRING2014_CSCI570_EXAM2

#183

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

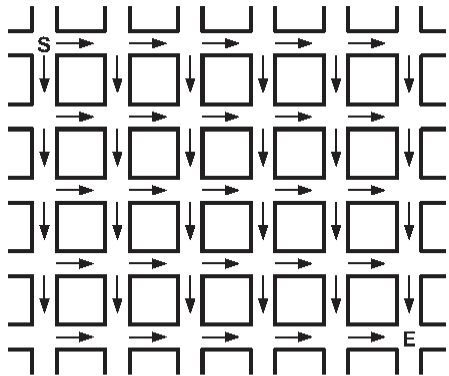


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



1D23EB70-C01C-40D4-85B6-69C6D240E6D2

SPRING2014_CSCI570_EXAM2

#183 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

6CDEBCDD-8CA1-49D8-B713-D755DED295DD

SPRING2014_CSCI570_EXAM2

#183

9 of 10



Additional Space



248204AC-3546-4ABE-9B2D-C9126BE250A5

SPRING2014_CSCI570_EXAM2

#183 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



23B3AC09-1052-4518-807A-7A677977DBE9

SPRING2014_CSCI570_EXAM2

#184 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

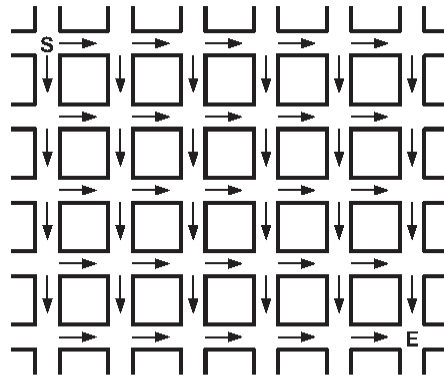


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



82C0C747-B851-4688-B682-D0636618FC92

SPRING2014_CSCI570_EXAM2

#184 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

0680164A-A78F-4D79-9A77-7BB4F7F9B8EF

SPRING2014_CSCI570_EXAM2

#184

9 of 10



Additional Space



65FD2346-6EEA-4C35-AD1A-4583E0BAAD3C

SPRING2014_CSCI570_EXAM2

#184 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



F5E60EDF-213C-4ED6-A255-C679038D01C2

SPRING2014_CSCI570_EXAM2

#185 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



7AA72B2F-91D3-42E7-ABA2-9B6EF36ABEE5

SPRING2014_CSCI570_EXAM2

#185 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

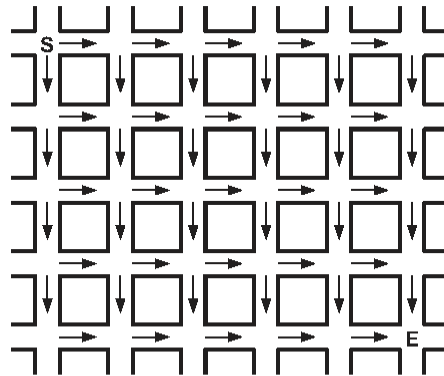


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



C4EB51EB-9D28-45CF-AC7A-50E16911EB56

SPRING2014_CSCI570_EXAM2

#185 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

B9DCA41B-C43E-4408-9281-93678D9734BD

SPRING2014_CSCI570_EXAM2

#185

9 of 10



Additional Space



831A5063-38EB-4237-A79A-C3562222AE1C

SPRING2014_CSCI570_EXAM2

#185 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



8393677D-3ECF-470A-BD10-F8EAF7797BD9

SPRING2014_CSCI570_EXAM2

#186 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



12882D94-52CB-42B0-BE65-C4713D1EED74

SPRING2014_CSCI570_EXAM2

#186 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

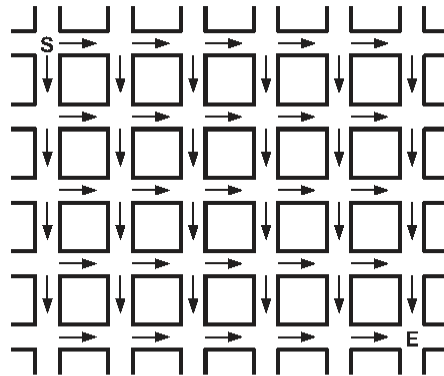


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



2A685AAB-E8BA-4A24-9C6D-55E4DBFD7C2F

SPRING2014_CSCI570_EXAM2

#186 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

DC05007E-0C49-45B1-94C7-06269D779060

SPRING2014_CSCI570_EXAM2

#186

9 of 10



Additional Space



184F9EC6-B261-460B-9934-BD0040756165

SPRING2014_CSCI570_EXAM2

#186 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



ACD3EB71-F13F-413F-8573-F8B248CD8BAF

SPRING2014_CSCI570_EXAM2

#187 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



B73F106A-59E6-4AB2-9357-A0D66969C7CF

SPRING2014_CSCI570_EXAM2

#187 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

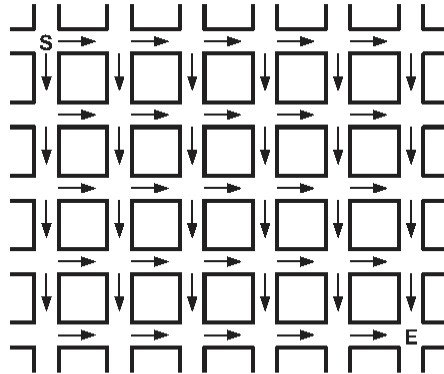


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



854F9351-CC83-4E0A-BFEB-6E20638C2B2F

SPRING2014_CSCI570_EXAM2

#187 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

07CCD5FE-417F-4209-8584-56777B003A21

SPRING2014_CSCI570_EXAM2

#187

9 of 10



Additional Space



3B3102A7-B161-490D-AA26-D9E6F94D774A

SPRING2014_CSCI570_EXAM2

#187 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



7893EDC2-81C8-4AE9-8A5A-29C1EEFB97C1

SPRING2014_CSCI570_EXAM2

#188

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

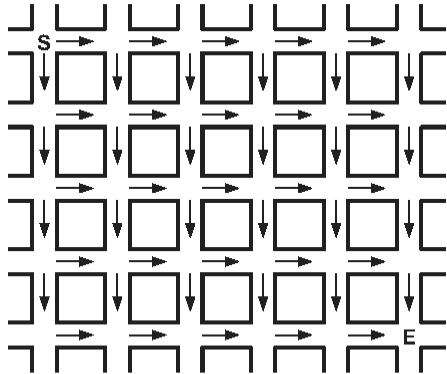


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



89D08BE4-B65F-43AF-9D28-C766CE674F78

SPRING2014_CSCI570_EXAM2

#188 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

F7B33CDD-1507-4D68-A1F4-1BCAEB6E85B5

SPRING2014_CSCI570_EXAM2

#188

9 of 10



Additional Space



8CF2FD5C-96FA-45D1-9097-4E450B786902

SPRING2014_CSCI570_EXAM2

#188 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



92B3244B-42FD-4999-91DD-AC898739269A

SPRING2014_CSCI570_EXAM2

#189 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



13FE0681-D23C-43DA-990C-B1CF49332A88

SPRING2014_CSCI570_EXAM2

#189

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

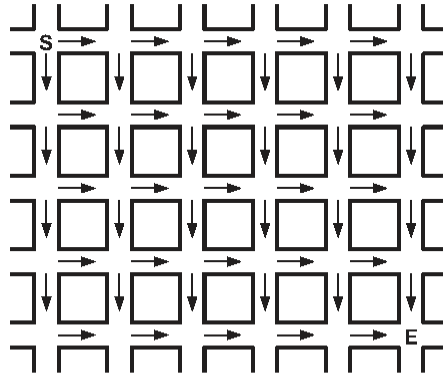


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

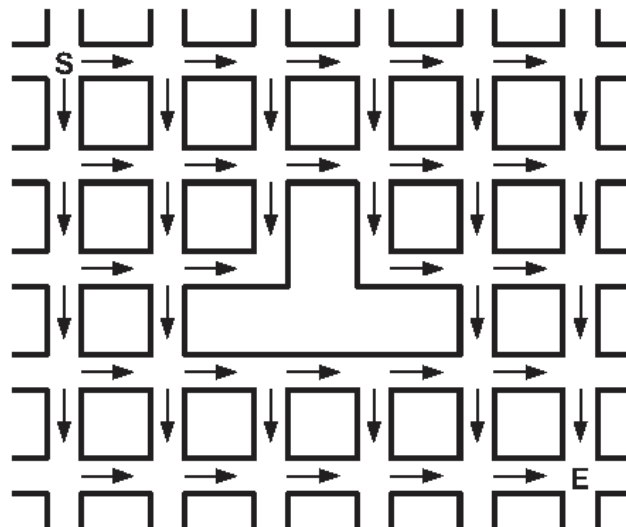


Figure B.



F14F10B5-35CE-4C9F-BF2C-00D712F0DA67

SPRING2014_CSCI570_EXAM2

#189 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

4B3A6914-3129-4505-AC75-17E818E3755C

SPRING2014_CSCI570_EXAM2

#189

9 of 10



Additional Space



CDD7FA39-B39A-4A6A-9079-9D008C5FDC1A

SPRING2014_CSCI570_EXAM2

#189 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



27417429-DA32-4374-9358-7EDDB5C49E1C

SPRING2014_CSCI570_EXAM2

#190 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



5) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

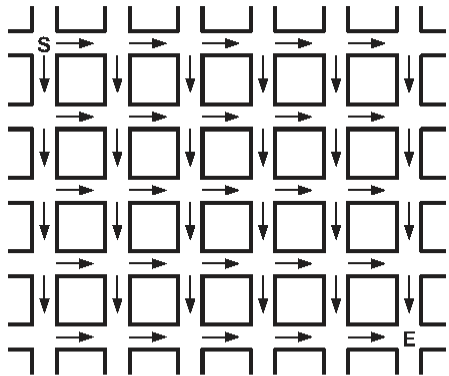


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



56D89264-FF6D-4D51-9889-661301728DF6

SPRING2014_CSCI570_EXAM2

#190 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

7F7BD239-D9EE-4B3F-BA0C-866DBBA24CB5

SPRING2014_CSCI570_EXAM2

#190

9 of 10



Additional Space



2B581FDC-0AB7-4A53-9EC0-2BE18FCBB0DF

SPRING2014_CSCI570_EXAM2

#190 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



5AEAC815-58C2-4823-9E5B-D6DD41E71980

SPRING2014_CSCI570_EXAM2

#191 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



2D3BD594-8AFD-4CCE-8B0D-B4C3EEDFC51B

SPRING2014_CSCI570_EXAM2

#191 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

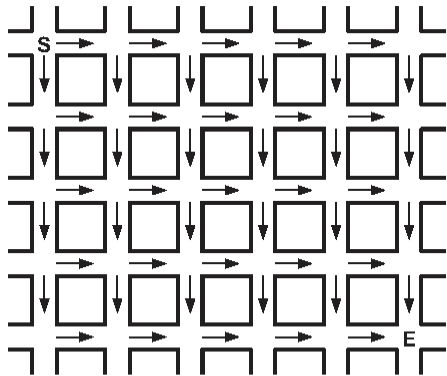


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

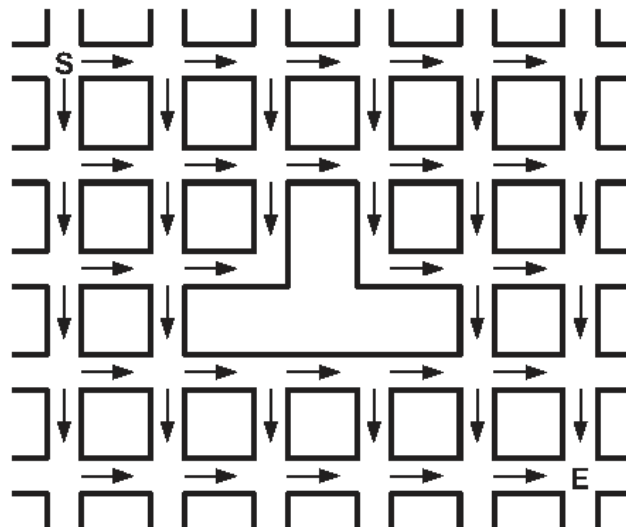


Figure B.



D89C7B6E-2E39-447A-893D-320DF7F09BAE

SPRING2014_CSCI570_EXAM2

#191 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

1F28D1E4-2FA3-41C0-95B1-BE5CA6C891E3

SPRING2014_CSCI570_EXAM2

#191

9 of 10



Additional Space



17374B8D-0969-45CE-85EC-E8D258A6AF86

SPRING2014_CSCI570_EXAM2

#191 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



862FF8EF-DB86-4408-8252-D537FE6BF1AF

SPRING2014_CSCI570_EXAM2

#192 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

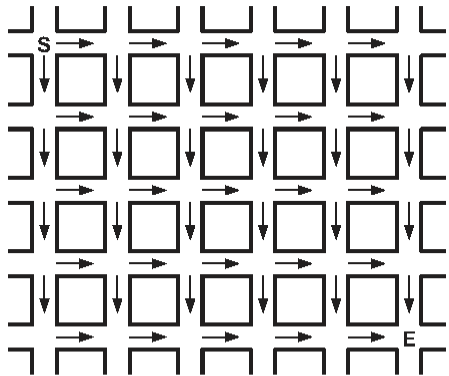


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



54EE3E86-CB44-465A-A11B-107E05DAED3D

SPRING2014_CSCI570_EXAM2

#192 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

00E49C68-052D-41E7-840D-A8BE264BF010

SPRING2014_CSCI570_EXAM2

#192 9 of 10



Additional Space



839D0638-542B-4479-B640-8C994BD833C8

SPRING2014_CSCI570_EXAM2

#192 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



AD565E36-3E61-426D-B14B-2F18406CE363

SPRING2014_CSCI570_EXAM2

#193 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



D0FA9602-4C89-4A52-9EF5-C07BC6F4CB14

SPRING2014_CSCI570_EXAM2

#193 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

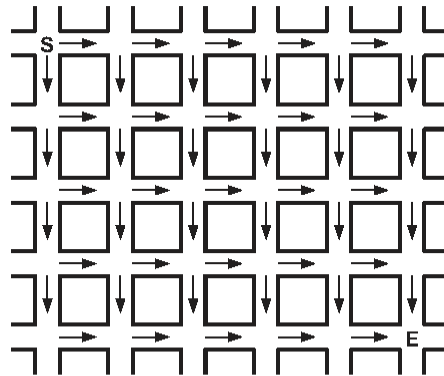


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

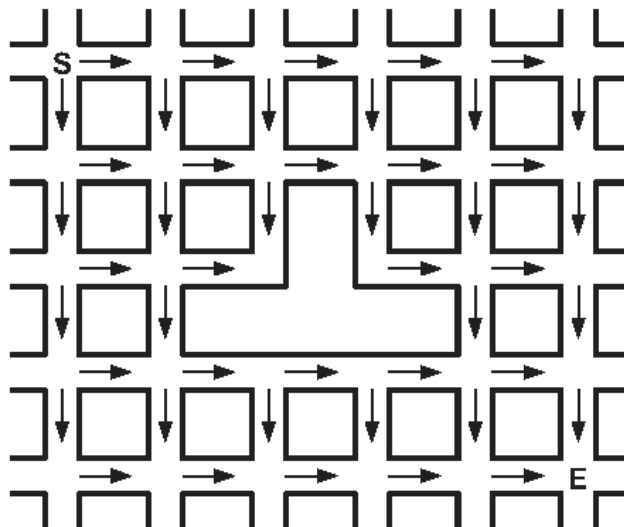


Figure B.



F88B5D60-E231-451B-B647-1F704E4A7F3B

SPRING2014_CSCI570_EXAM2

#193 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

5991AC91-65B7-47D5-9A61-49490C3BA4E6

SPRING2014_CSCI570_EXAM2

#193

9 of 10



Additional Space



58CD5B1C-9A4C-47C2-BEED-C33AABE9782F

SPRING2014_CSCI570_EXAM2

#193 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



E3966148-0C15-4658-A498-A31A6840EEA3

SPRING2014_CSCI570_EXAM2

#194 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

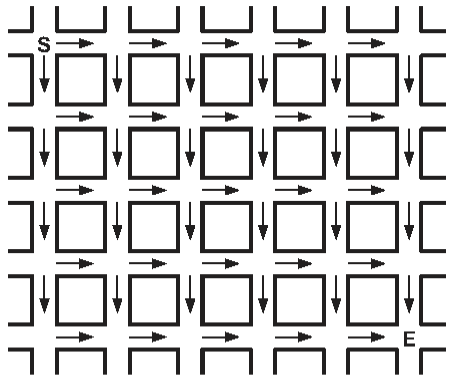


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



A9E686A3-019C-48D0-AE86-CFC40BB83DC3

SPRING2014_CSCI570_EXAM2

#194 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

20046456-AD84-4F6A-939D-B5CA7613FF36

SPRING2014_CSCI570_EXAM2

#194

9 of 10



Additional Space



5811552F-7464-46E6-9366-364E7C4EBB03

SPRING2014_CSCI570_EXAM2

#194 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



7C28FC4D-4749-4C25-9BED-ECB62FFAF144

SPRING2014_CSCI570_EXAM2

#195 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



FE7D1F98-195E-4F5D-8075-A70CDB6E0BC5

SPRING2014_CSCI570_EXAM2

#195 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

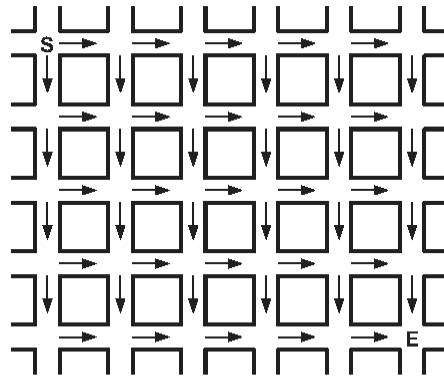


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



EE03E25E-A1A1-4284-9392-4804A992E51B

SPRING2014_CSCI570_EXAM2

#195 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

55959D05-7B38-4A87-8DC0-3493B1F4C758

SPRING2014_CSCI570_EXAM2

#195

9 of 10



Additional Space



78C87127-DD1E-46C2-930C-872C2386E39F

SPRING2014_CSCI570_EXAM2

#195 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



4BF06705-DD92-44EE-AC2D-497092EEAAED

SPRING2014_CSCI570_EXAM2

#196 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



D526F986-D377-48E5-8896-D65E44D531F5

SPRING2014_CSCI570_EXAM2

#196

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required)
(15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

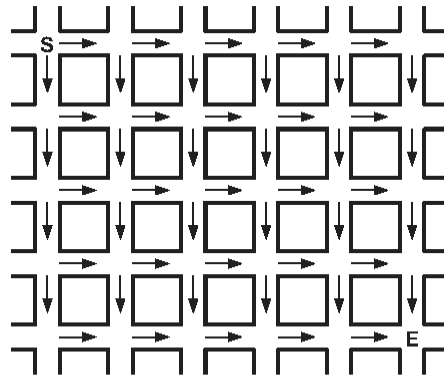


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



AD224EBE-4213-4E16-9674-76FB4BDB96A3

SPRING2014_CSCI570_EXAM2

#196 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

71AC2794-5B1E-4CD9-A6C5-89F7ADA0E7E3

SPRING2014_CSCI570_EXAM2

#196 9 of 10



Additional Space



606C18BE-35E0-4728-AA10-C0B622017AEA

SPRING2014_CSCI570_EXAM2

#196 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



50B6FD67-3183-4FA6-A51B-BF427894E960

SPRING2014_CSCI570_EXAM2

#197 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



6071BEE8-BAFD-4945-A7DD-7F7262FE7106

SPRING2014_CSCI570_EXAM2

#197 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

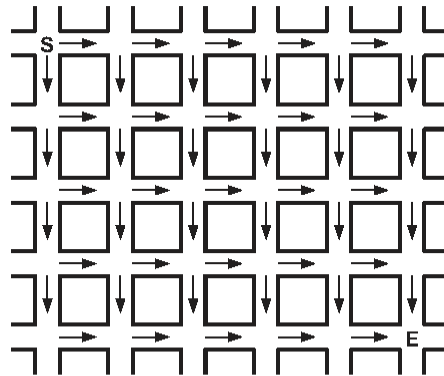


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

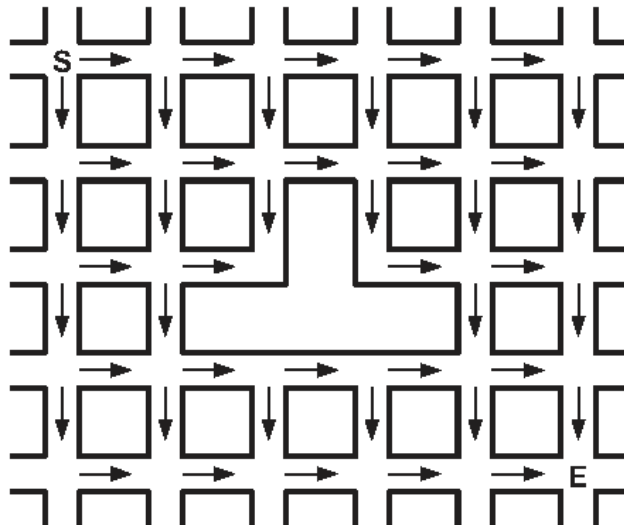


Figure B.



FF28A233-FDBA-4F2A-A3D6-7E85D96F4F44

SPRING2014_CSCI570_EXAM2

#197 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

47A89E3A-202F-42D3-8127-CED79411BF5A

SPRING2014_CSCI570_EXAM2

#197

9 of 10



Additional Space



CAB3E172-5E85-4C6B-88BD-99F905EED926

SPRING2014_CSCI570_EXAM2

#197 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



6E9AB414-CEA2-4999-A590-1458F4BDAAF5

SPRING2014_CSCI570_EXAM2

#198 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



B65BC7A1-0F77-4E65-A0C4-5A4AC0581443

SPRING2014_CSCI570_EXAM2

#198

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

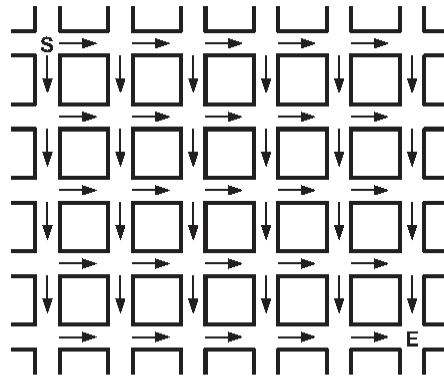


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

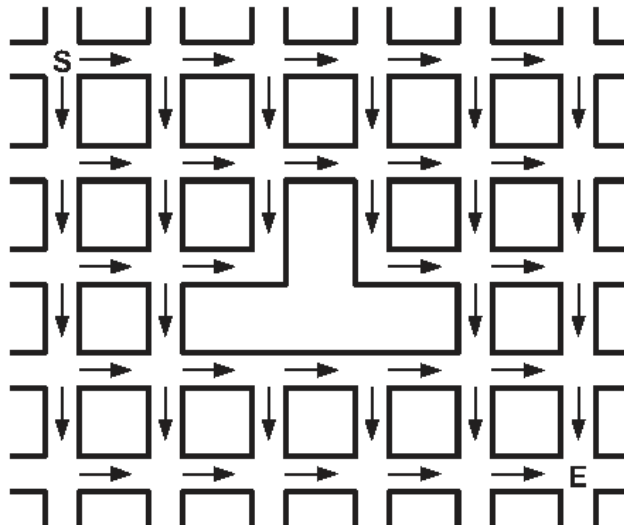


Figure B.



07EE00B8-51E4-45B9-8B9B-05350D8ECC4B

SPRING2014_CSCI570_EXAM2

#198 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

36917CBB-14F0-4DBB-8F03-0287F8945E0D

SPRING2014_CSCI570_EXAM2

#198

9 of 10



Additional Space



3C3FCAA6-8D53-4FB2-8F44-46C97AB63532

SPRING2014_CSCI570_EXAM2

#198 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



E56CA334-9CCF-4739-8752-D4158A978C0E

SPRING2014_CSCI570_EXAM2

#199

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



3B1DF7B3-12C3-4EE8-9E42-6DE499708C89

SPRING2014_CSCI570_EXAM2

#199

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

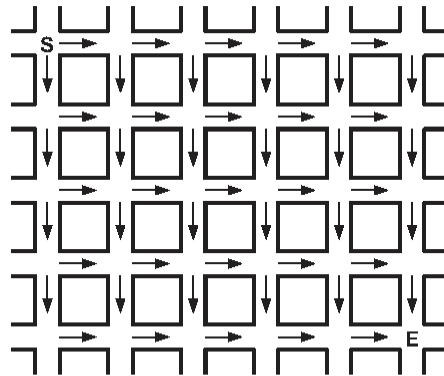


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



ABFE0551-7C45-4FB9-A511-F264671A6E3D

SPRING2014_CSCI570_EXAM2

#199 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

E9588D1F-E26E-479C-9D29-CA48DFD610FE

SPRING2014_CSCI570_EXAM2

#199

9 of 10



Additional Space



993F7EA5-938A-4598-8E0D-AA2CB544B75C

SPRING2014_CSCI570_EXAM2

#199 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



3BAC6DD5-8272-4A8E-B725-EE913873EB9C

SPRING2014_CSCI570_EXAM2

#200 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



DB552800-9FCF-48FD-BE19-CDEA6AC553C6

SPRING2014_CSCI570_EXAM2

#200 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

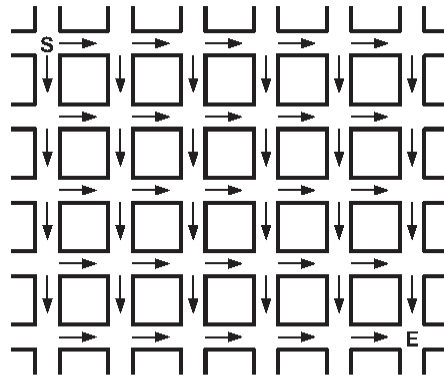


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

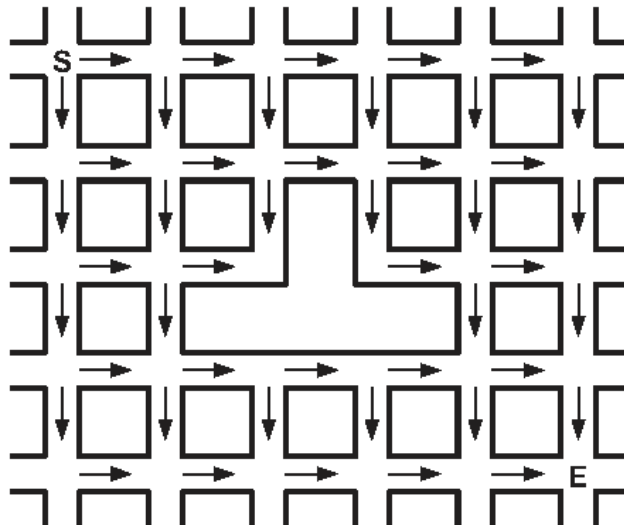


Figure B.



13580A6C-992A-4AEB-BDF9-D7D097393322

SPRING2014_CSCI570_EXAM2

#200 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

85CA236C-C09A-4F77-B0C0-3E938B79D230

SPRING2014_CSCI570_EXAM2

#200

9 of 10



Additional Space



C1455FCE-5D63-4257-B7A1-4654FCD2A586

SPRING2014_CSCI570_EXAM2

#200 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



9F68AE41-1480-4D27-BE59-49648838E0E9

SPRING2014_CSCI570_EXAM2

#201 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



6D6CF719-387D-4FF6-85E9-9514C67F309C

SPRING2014_CSCI570_EXAM2

#201 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



507897AC-9B22-4AF0-B6A0-6F181397422C

SPRING2014_CSCI570_EXAM2

#201 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

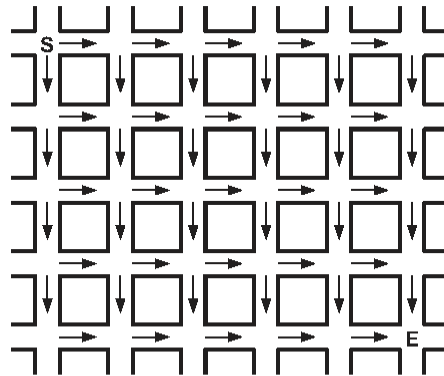


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



93ABFABD-B00F-4BC1-91B5-E8C991175C03

SPRING2014_CSCI570_EXAM2

#201 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

AFEA8FA8-5CFB-477C-BA7B-60B6D7341D5F

SPRING2014_CSCI570_EXAM2

#201 9 of 10



Additional Space



D083D768-C342-482A-86D4-E6A16213345A

SPRING2014_CSCI570_EXAM2

#201 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



D395DDFF-3A54-4CB1-ADAE-65187ABB5989

SPRING2014_CSCI570_EXAM2

#202 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



C4E64E23-1B91-4C69-9959-508E6D02150B

SPRING2014_CSCI570_EXAM2

#202 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

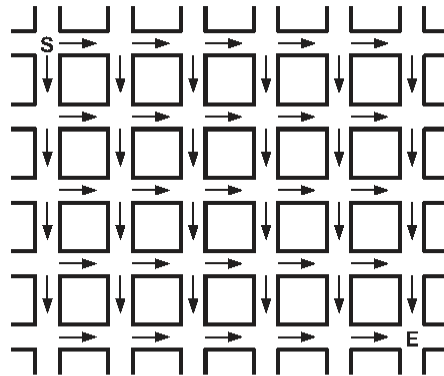


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



E18E3A78-7424-45E0-8EE1-4A4CA5E4419A

SPRING2014_CSCI570_EXAM2

#202 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

F8062697-99AC-43A0-AFD5-DB5E1ADBB48D

SPRING2014_CSCI570_EXAM2

#202 9 of 10



Additional Space



3961E91C-B1EE-43B2-B9B0-6F8D2D961730

SPRING2014_CSCI570_EXAM2

#202 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



08BA7150-4162-4D42-8973-53813BF3EA7A

SPRING2014_CSCI570_EXAM2

#203 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



09F4DB76-C664-4C81-8F6C-B31A96F52BDD

SPRING2014_CSCI570_EXAM2

#203 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

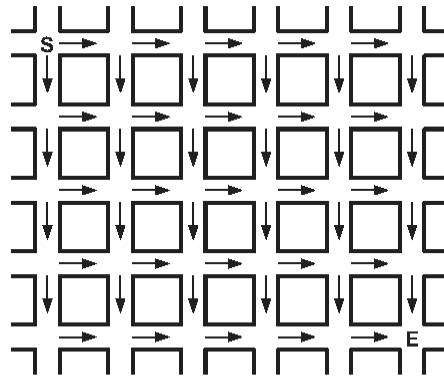


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



122006B4-29D2-4352-A86D-9CAC2B879C2B

SPRING2014_CSCI570_EXAM2

#203 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

74CD249B-BC02-414A-8184-F08F4BEBB046

SPRING2014_CSCI570_EXAM2

#203

9 of 10



Additional Space



DA7C320C-FFBD-43EA-AB4C-5E9E6E9EC0D0

SPRING2014_CSCI570_EXAM2

#203 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



18DC4AAA-7DA9-4BFD-BDF2-42469B8C1BBA

SPRING2014_CSCI570_EXAM2

#204 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

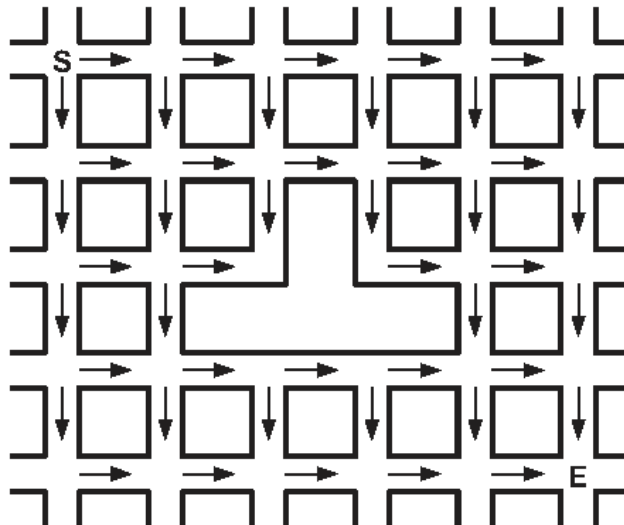


Figure B.



AF5C74B9-734E-4203-9E7B-0B29E341E8DC

SPRING2014_CSCI570_EXAM2

#204 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

E978E7C1-27BB-4F99-98F3-0E6DEE1B9ACD

SPRING2014_CSCI570_EXAM2

#204

9 of 10



Additional Space



B17D601B-C944-4E73-8069-80C378756D75

SPRING2014_CSCI570_EXAM2

#204 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



889704A7-9D31-47E5-9C4A-EBA74138BA83

SPRING2014_CSCI570_EXAM2

#205 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

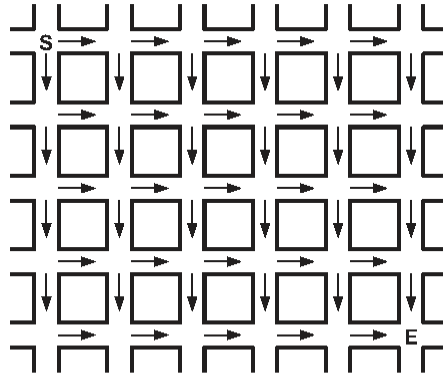


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



2A5AC59D-AE1F-4992-947F-91AA7CD011FF

SPRING2014_CSCI570_EXAM2

#205 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

49903920-E6D2-4C28-A903-DFA0E5A4374E

SPRING2014_CSCI570_EXAM2

#205

9 of 10



Additional Space



7555A591-2BBA-45F9-BAD4-E473C25CFA5A

SPRING2014_CSCI570_EXAM2

#205 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



95F6E71A-FF01-49DC-8B65-8515653956B4

SPRING2014_CSCI570_EXAM2

#206 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

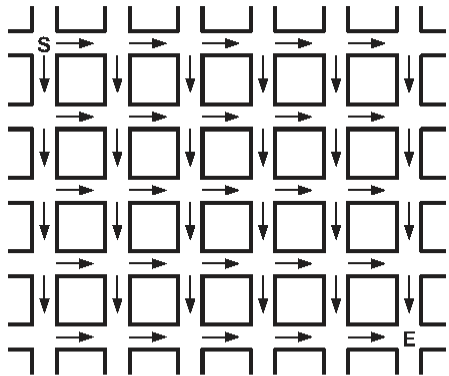


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



D845B76A-91EF-4C25-9EF0-82C1E648FF81

SPRING2014_CSCI570_EXAM2

#206 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

485B494C-26D8-42B0-9ECC-C57E92B93C09

SPRING2014_CSCI570_EXAM2

#206

9 of 10



Additional Space



63EBEE8F-EC20-444F-9CFB-EC6C66B23E62

SPRING2014_CSCI570_EXAM2

#206 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



37B6A955-75C7-4324-98D9-D47144A14F26

SPRING2014_CSCI570_EXAM2

#207 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



96077845-5BC2-45D4-8A34-3D5B4A23958A

SPRING2014_CSCI570_EXAM2

#207 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

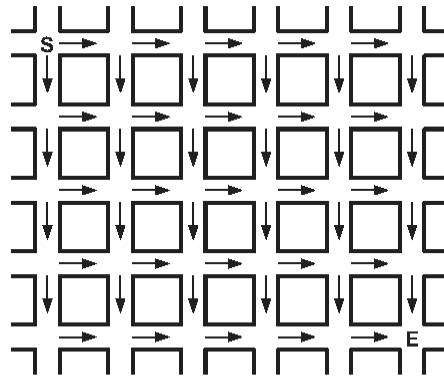


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



DCC4DAC9-CC79-4411-A7E6-C632A50EE916

SPRING2014_CSCI570_EXAM2

#207 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

E74C1D06-260A-47DA-AC03-9662BBFC9A84

SPRING2014_CSCI570_EXAM2

#207

9 of 10



Additional Space



A2272546-6FEB-4358-A827-56D0E6DA8DD1

SPRING2014_CSCI570_EXAM2

#207 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



12D8298A-6750-4FA0-B3E0-D34A59190E6E

SPRING2014_CSCI570_EXAM2

#208 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

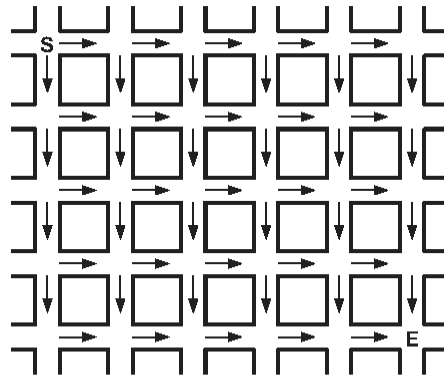


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

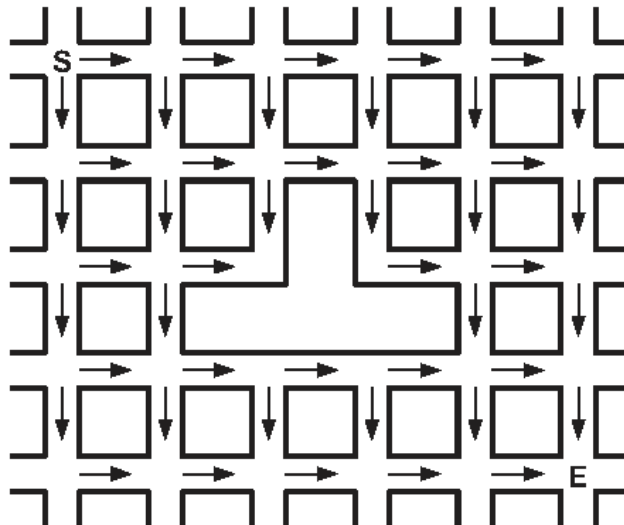


Figure B.



D44F2406-77B2-4002-804E-F3410C9859EC

SPRING2014_CSCI570_EXAM2

#208 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

B6342784-1427-40FC-9B8F-A305197A40FA

SPRING2014_CSCI570_EXAM2

#208

9 of 10



Additional Space



F5A9C985-2CA0-4AD9-83CE-EB2CF1CC0DDF

SPRING2014_CSCI570_EXAM2

#208 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



94ED3BE3-B862-48D3-847D-159BB127FA7B

SPRING2014_CSCI570_EXAM2

#209 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



7E179615-8272-435D-8B88-238EA8180CE8

SPRING2014_CSCI570_EXAM2

#209 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



92BD571F-3006-440B-B6BF-6129AAA1A6B3

SPRING2014_CSCI570_EXAM2

#209 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

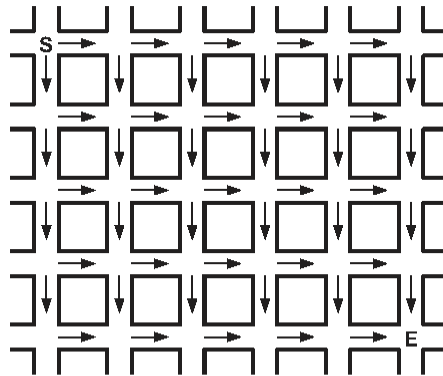


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



C2D32748-4892-49BA-A511-385AD9D9CE2D

SPRING2014_CSCI570_EXAM2

#209 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

92801E4A-1434-462E-B4AE-DB0ADDF55466

SPRING2014_CSCI570_EXAM2

#209

9 of 10



Additional Space



5B9D277B-67A7-4CDB-A1E9-A609AB028929

SPRING2014_CSCI570_EXAM2

#209 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



799AFE25-45D7-45B4-ACB3-FDA1FA32292E

SPRING2014_CSCI570_EXAM2

#210 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



12A7ADFB-7A00-4552-A5CC-86E117F3F5C0

SPRING2014_CSCI570_EXAM2

#210 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

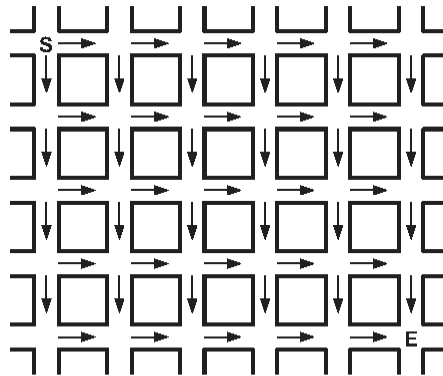


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

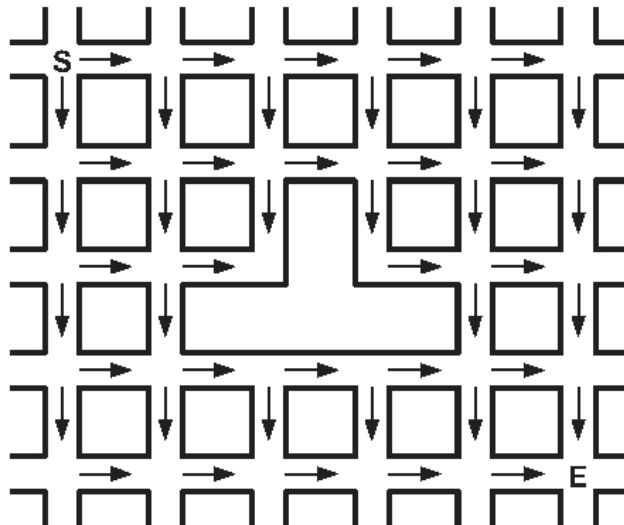


Figure B.



E99595FB-F7B5-4793-9673-14757AE795AF

SPRING2014_CSCI570_EXAM2

#210 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

16E38F41-736B-4229-B352-BFDD771A027F

SPRING2014_CSCI570_EXAM2

#210

9 of 10



Additional Space



861D9C76-2456-48CD-861E-223E696B7206

SPRING2014_CSCI570_EXAM2

#210 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



FE5A263A-F491-42F5-B201-C806B03224FE

SPRING2014_CSCI570_EXAM2

#211 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



EEA4B81B-1A77-45A3-A95F-B9E9D8A1AEC3

SPRING2014_CSCI570_EXAM2

#211 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

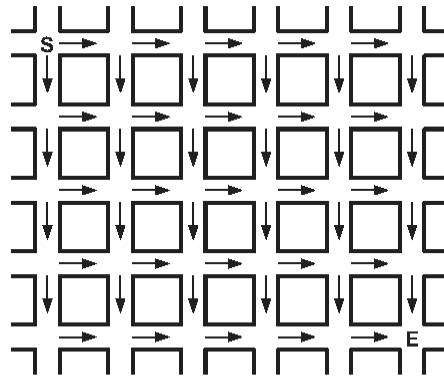


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

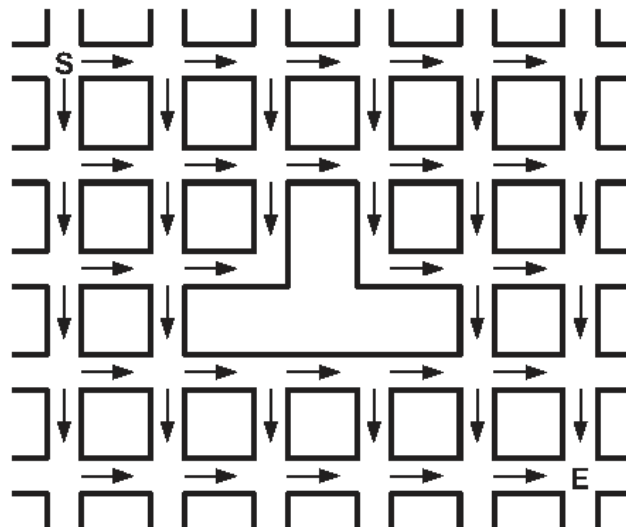


Figure B.



B45481E2-1CE6-4A29-813A-221A0FD839BD

SPRING2014_CSCI570_EXAM2

#211 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

67E5A350-7BA9-47B4-B394-C17B7C2B0073

SPRING2014_CSCI570_EXAM2

#211

9 of 10



Additional Space



08B8DA14-93FD-461A-8CAB-B98FFD83C627

SPRING2014_CSCI570_EXAM2

#211 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



7B2D1F09-2C48-4D4C-9C4A-71DB70F7822D

SPRING2014_CSCI570_EXAM2

#212 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



2D15F320-B902-4204-9267-F70A14D85D46

SPRING2014_CSCI570_EXAM2

#212 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.

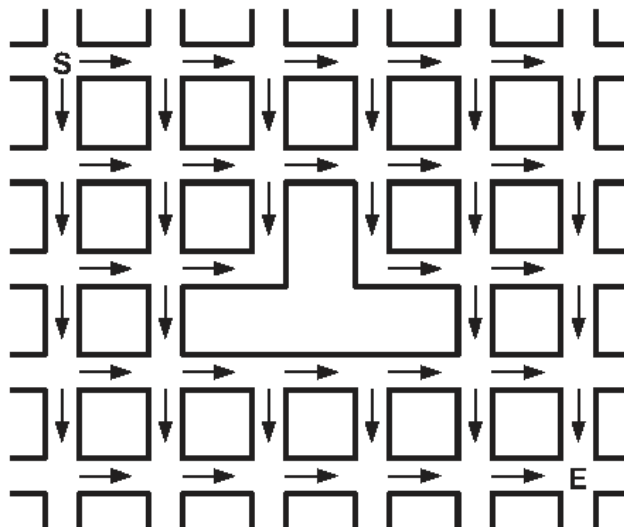


Figure B.



10BA8B9D-8CF8-4DD9-8CC8-346AD4FEE1C3

SPRING2014_CSCI570_EXAM2

#212 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

553AB6A8-D713-4258-9E1E-8787B6DAD095

SPRING2014_CSCI570_EXAM2

#212

9 of 10



Additional Space



A0DF95FD-50DB-4738-98AC-F18307B34470

SPRING2014_CSCI570_EXAM2

#212 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



305F93DF-B465-4C02-9A03-FFDC8DA76A0F

SPRING2014_CSCI570_EXAM2

#213 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

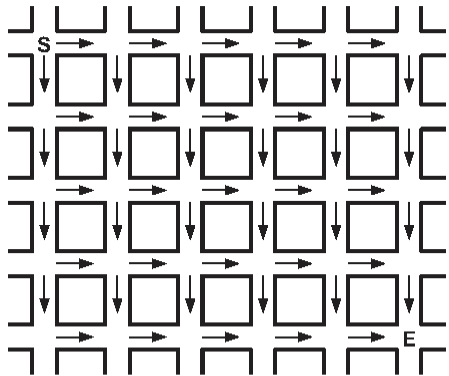


Figure A.

-

Figure B.



9EBAA070-6F50-47B8-BD1E-A5D4FBC83522

SPRING2014_CSCI570_EXAM2

#213 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

45EB6713-870D-4D68-B17F-CDB1615B472D

SPRING2014_CSCI570_EXAM2

#213

9 of 10



Additional Space



C879F7E2-E6A7-41CD-B273-BF26F54AD9E6

SPRING2014_CSCI570_EXAM2

#213 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



063587AE-8E55-45FF-8438-782D920DD96A

SPRING2014_CSCI570_EXAM2

#214 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



67761509-5DC0-47F2-BE1C-795BBA05C5EC

SPRING2014_CSCI570_EXAM2

#214 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



96EBD795-9935-4488-B70A-7D50F7A907C8

SPRING2014_CSCI570_EXAM2

#214

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

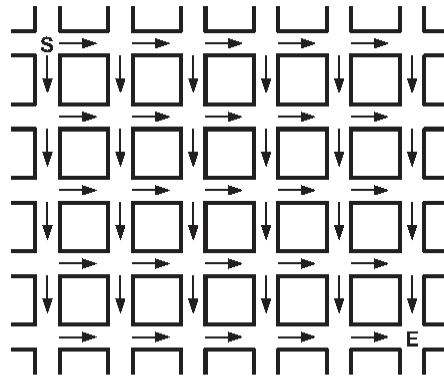


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

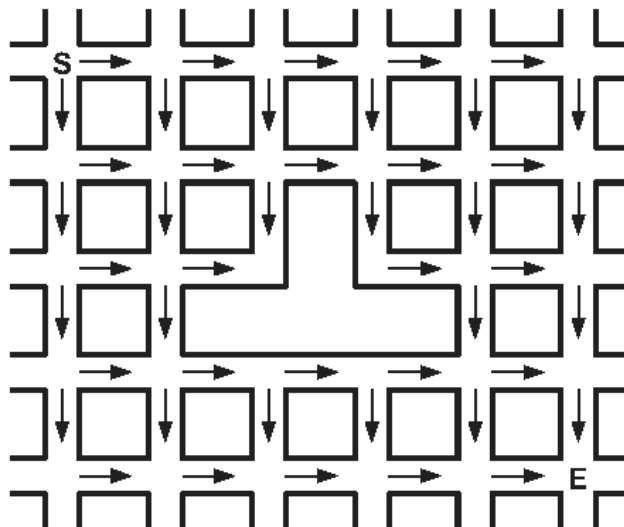


Figure B.



BC969364-A47B-4555-8F0A-877BA53B2DE5

SPRING2014_CSCI570_EXAM2

#214 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

7E6408D8-18CC-4B5E-AF93-651CD48CDC3B

SPRING2014_CSCI570_EXAM2

#214

9 of 10



Additional Space



636D1315-BDFA-490D-9D90-154DF7E5F82B

SPRING2014_CSCI570_EXAM2

#214 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



5F4430B1-1540-4E66-BF0E-64417061D2D6

SPRING2014_CSCI570_EXAM2

#215 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

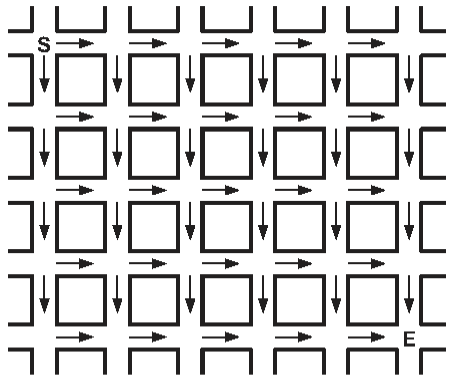


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

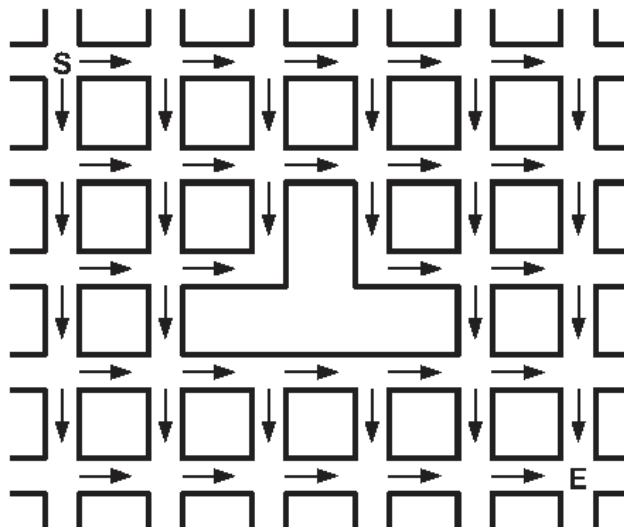


Figure B.



DF737C25-4340-4E98-B72F-9FF8E68AD2CC

SPRING2014_CSCI570_EXAM2

#215 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

0E85E664-4447-4AA8-A1DE-8C4E25AAD91F

SPRING2014_CSCI570_EXAM2

#215

9 of 10



Additional Space



B5A360D1-6931-4E6D-9E5E-ED9504C6CA3F

SPRING2014_CSCI570_EXAM2

#215 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



F934FE7C-5019-4A80-995D-9A67DA2CA399

SPRING2014_CSCI570_EXAM2

#216

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

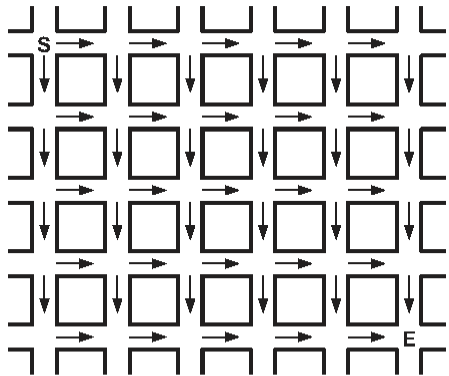


Figure A.

-

Figure B.



748CDA37-EE26-463D-ADE6-68718D3F9419

SPRING2014_CSCI570_EXAM2

#216 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

1032C837-9315-49C7-A923-65D85CEECEB3

SPRING2014_CSCI570_EXAM2

#216

9 of 10



Additional Space



175BA2D8-C5A1-4C8E-8FB7-6B006546F523

SPRING2014_CSCI570_EXAM2

#216 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



C25A08FB-2C22-4763-A62B-ADEABC69CCE0

SPRING2014_CSCI570_EXAM2

#217 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

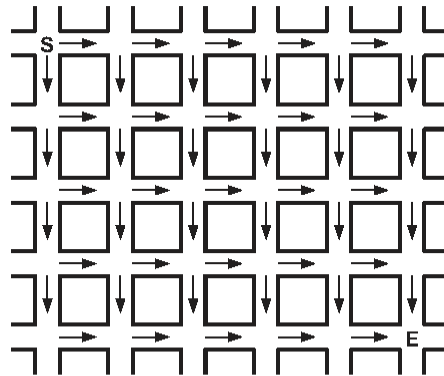


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



5D34AABB-C642-4952-961A-63303D74BC2E

SPRING2014_CSCI570_EXAM2

#217 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

A525048B-E27C-4D68-9BC2-A4562730D1EB

SPRING2014_CSCI570_EXAM2

#217

9 of 10



Additional Space



DF9D56C7-51F8-4F5C-BAE3-E7081F253693

SPRING2014_CSCI570_EXAM2

#217 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



65B23121-7587-4A91-811B-0599E379AC83

SPRING2014_CSCI570_EXAM2

#218 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



F4DD64C0-9357-4602-BA40-9378698522DC

SPRING2014_CSCI570_EXAM2

#218

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



8AF8C7AA-6BF0-49F4-9B95-A56306DBF7E2

SPRING2014_CSCI570_EXAM2

#218

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

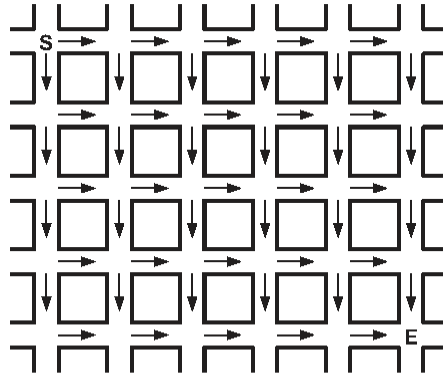


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

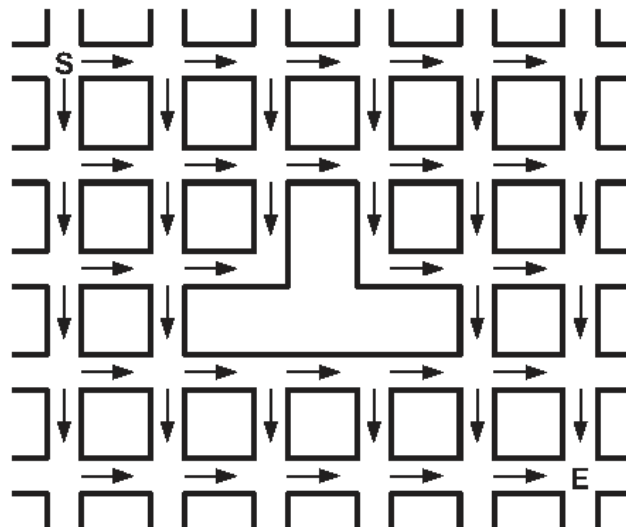


Figure B.



7E165386-509F-4154-8FE9-93ED74019F1B

SPRING2014_CSCI570_EXAM2

#218 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

62DB4B49-213A-4474-B214-5C73813178E9

SPRING2014_CSCI570_EXAM2

#218

9 of 10



Additional Space



927FF76F-9150-42EC-A66C-8BBC2CCB133B

SPRING2014_CSCI570_EXAM2

#218 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



7A533FFE-70DA-4D7D-9C3E-45DA42C50034

SPRING2014_CSCI570_EXAM2

#219

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

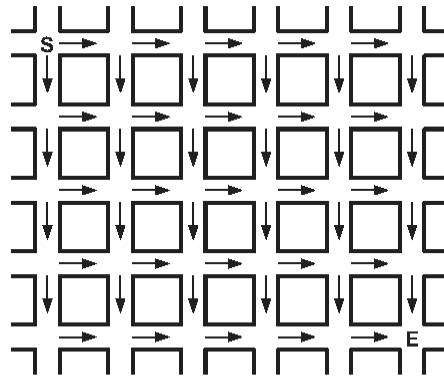


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

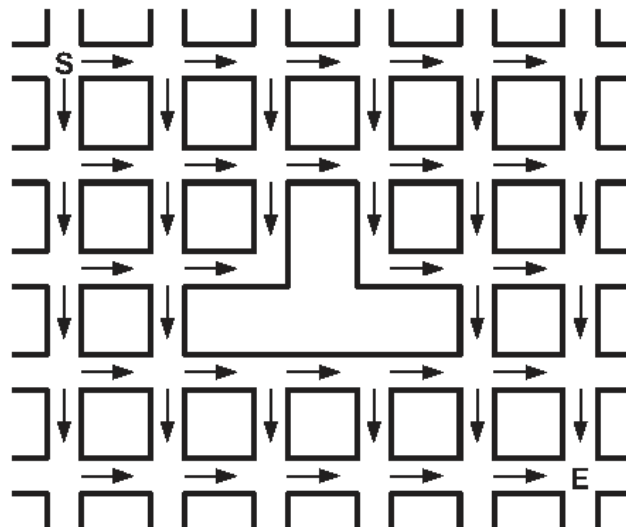


Figure B.



0B4090C1-C039-4DB6-AA9D-9EA3FDBA68F9

SPRING2014_CSCI570_EXAM2

#219 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

96EC278B-E049-4F9D-B287-0CA76549D086

SPRING2014_CSCI570_EXAM2

#219

9 of 10



Additional Space



E17F154E-3D85-4A85-82F7-B07E4F4FBF8C

SPRING2014_CSCI570_EXAM2

#219 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



860D4F61-491E-45A6-84BF-21608F7F68D5

SPRING2014_CSCI570_EXAM2

#220 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

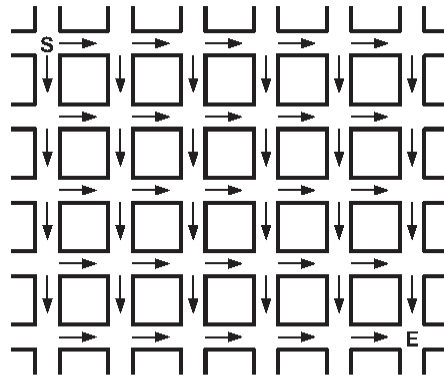


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



5695D9D0-C682-4DD0-B0E4-F4903529E444

SPRING2014_CSCI570_EXAM2

#220 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

75A53787-E996-4868-85E8-E96F5F6FE004

SPRING2014_CSCI570_EXAM2

#220

9 of 10



Additional Space



1DFE3A37-C9DD-43FB-B982-66B6AD8DBD8B

SPRING2014_CSCI570_EXAM2

#220 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



68BCF587-D968-4400-A3C8-5B368FE4C516

SPRING2014_CSCI570_EXAM2

#221 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



BB2761D3-52C4-47BD-8F3C-2633634F9C39

SPRING2014_CSCI570_EXAM2

#221 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



7D8FF45F-B437-4602-86A2-73493DCDFB09

SPRING2014_CSCI570_EXAM2

#221 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

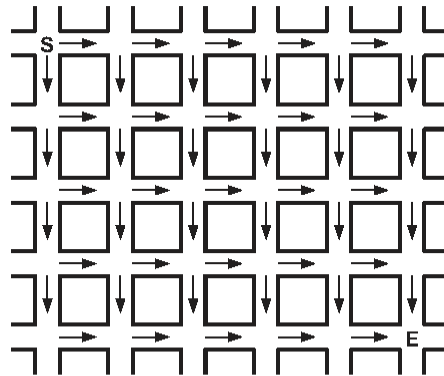


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\cdot, \cdot)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\cdot, \cdot)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

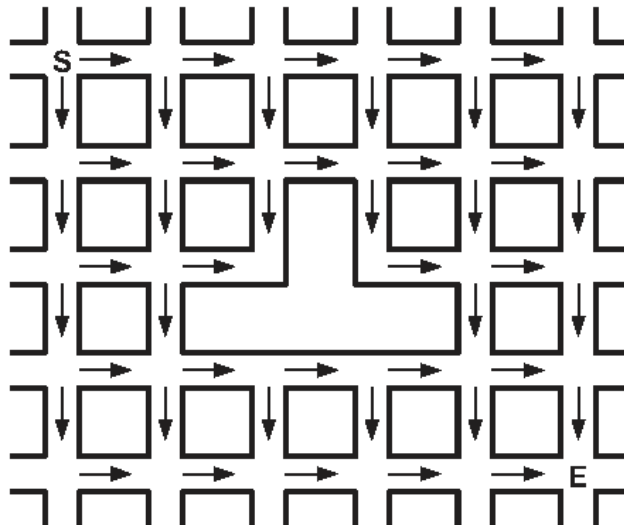


Figure B.



AC2E6998-1BEF-407F-8935-7B26AED76398

SPRING2014_CSCI570_EXAM2

#221 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

780AE985-1796-4485-8441-C996BDD4DA82

SPRING2014_CSCI570_EXAM2

#221

9 of 10



Additional Space



3DA527DE-A6A4-49B4-881E-7F95996C8DE3

SPRING2014_CSCI570_EXAM2

#221 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



23CD5D97-229E-4831-9225-D24BC5C4B057

SPRING2014_CSCI570_EXAM2

#222 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



9FAD48E2-D857-4183-9DC9-8F969B6BEA34

SPRING2014_CSCI570_EXAM2

#222 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



AF6E3F79-090E-4E57-8D0E-E74CC7AFBC71

SPRING2014_CSCI570_EXAM2

#222 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

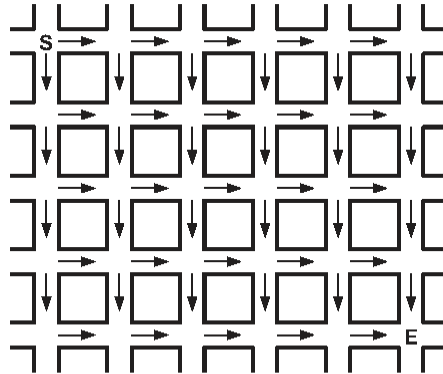


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

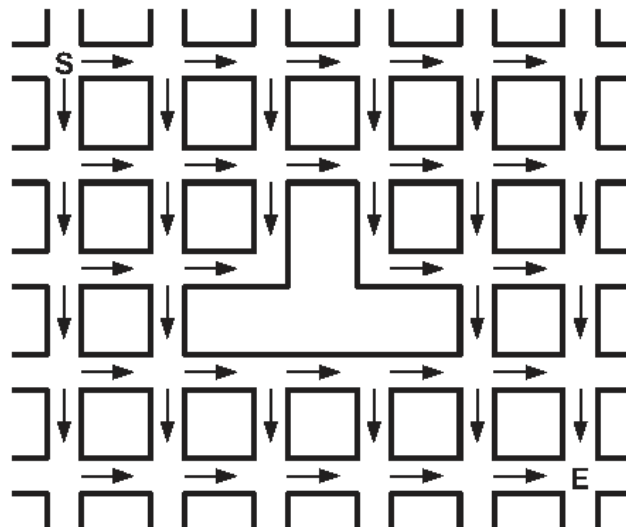


Figure B.



79B2FD54-2444-43E0-A344-9B61EEAF9AA3

SPRING2014_CSCI570_EXAM2

#222 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

73FA2E25-0A49-4377-9AF5-C231FB77EE51

SPRING2014_CSCI570_EXAM2

#222

9 of 10



Additional Space



1C6D2BD4-8033-4764-97EA-9EC48B5E10D3

SPRING2014_CSCI570_EXAM2

#222 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



358E3E4F-95F8-402A-914C-C7ABC7040916

SPRING2014_CSCI570_EXAM2

#223 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



99ACF6EC-FA64-4A64-ADDD-0F51996ABEEB

SPRING2014_CSCI570_EXAM2

#223 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

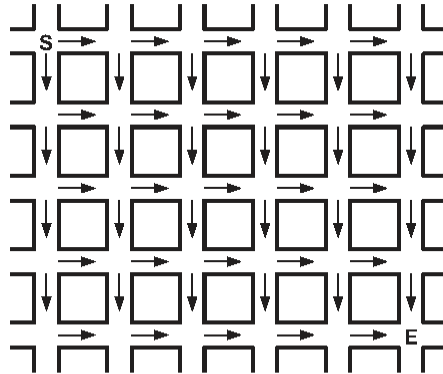


Figure A.

-
- A diagram of a square lattice with a central defect. The lattice is 6x6. A central 2x2 area is missing, replaced by a single large square. Arrows indicate flow: horizontal arrows point right, vertical arrows point down. The top-left corner is labeled 'S' and the bottom-right corner is labeled 'E'.

Figure B.



865601E1-D440-40A1-B6A7-B1647E761AE9

SPRING2014_CSCI570_EXAM2

#223 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

6B7D4CC0-4A29-4DB2-BF82-89160634F26D

SPRING2014_CSCI570_EXAM2

#223

9 of 10



Additional Space



5D52D6B1-AC03-42EF-972B-4939C276320B

SPRING2014_CSCI570_EXAM2

#223 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



25FF06BF-BFEE-4149-9894-5F43F48B6E60

SPRING2014_CSCI570_EXAM2

#224 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



788C4902-0ADE-4737-9C89-7F0D4F5503A9

SPRING2014_CSCI570_EXAM2

#224

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

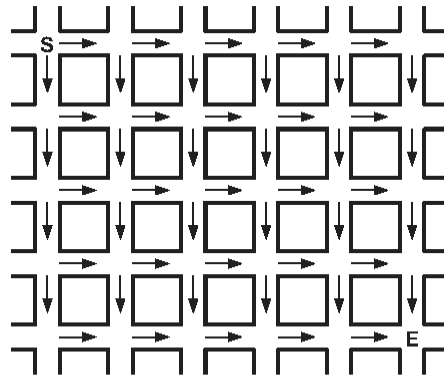


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

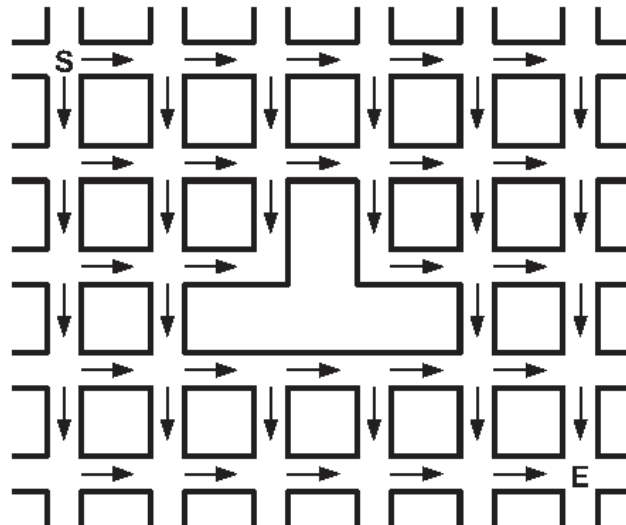


Figure B.



039E282B-3263-41BC-BC03-0DC310A9952A

SPRING2014_CSCI570_EXAM2

#224 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

A9623261-8A4C-49C4-A882-1F9F70AFD7AA

SPRING2014_CSCI570_EXAM2

#224

9 of 10



Additional Space



76BE01E0-8546-4673-B059-DD3923ACEAD5

SPRING2014_CSCI570_EXAM2

#224 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



53CBD2D8-9E7B-4312-B352-32B82E78FE06

SPRING2014_CSCI570_EXAM2

#225 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



7525929A-892D-44B6-8F25-8620A9196AFA

SPRING2014_CSCI570_EXAM2

#225

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

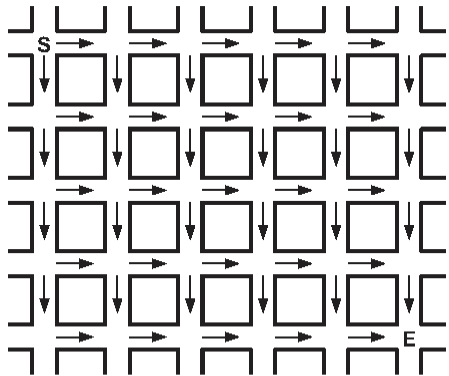


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



4504750E-F125-40A5-8611-37ED5E0795DE

SPRING2014_CSCI570_EXAM2

#225 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

AFABB089-2575-4DC7-B741-8B6CE232B3DD

SPRING2014_CSCI570_EXAM2

#225

9 of 10



Additional Space



2B2A2DE6-8C7E-448F-A715-5AC5DA933A0A

SPRING2014_CSCI570_EXAM2

#225 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



014EF756-777C-4295-B27C-6B62C09D7783

SPRING2014_CSCI570_EXAM2

#226

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

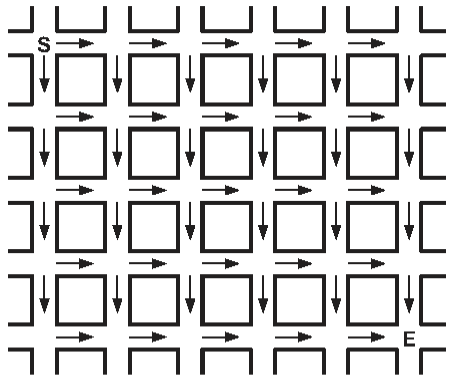


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



55F02555-AD2B-4603-BC98-290D1E690A90

SPRING2014_CSCI570_EXAM2

#226 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

CD218889-7769-40E4-8EAE-F2FA06E446EE

SPRING2014_CSCI570_EXAM2

#226

9 of 10



Additional Space



CD597E3A-378C-4A85-A8E6-78D332946CF6

SPRING2014_CSCI570_EXAM2

#226 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



2FF5B37D-3A27-4E65-A378-4DED89A6B9CE

SPRING2014_CSCI570_EXAM2

#227 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



F20181D0-A5A1-448A-96A0-7008591BBC46

SPRING2014_CSCI570_EXAM2

#227 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

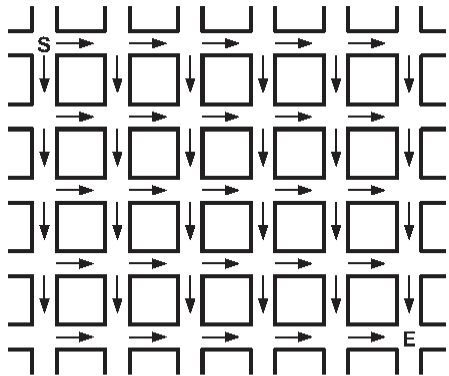


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

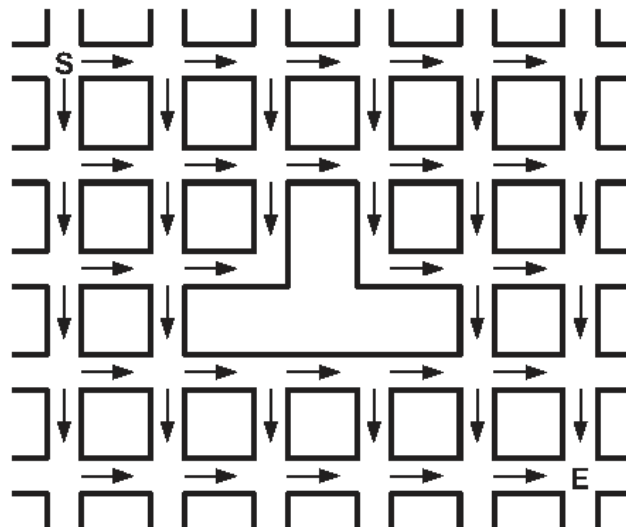


Figure B.



165FBB37-451D-4F66-9AE2-22EC7C504277

SPRING2014_CSCI570_EXAM2

#227 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

39D388C9-4701-4A54-9060-81F7E0882FF4

SPRING2014_CSCI570_EXAM2

#227

9 of 10



Additional Space



52F69A33-9535-407F-B41E-7859FE8E70CC

SPRING2014_CSCI570_EXAM2

#227 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



8935C054-2E80-4D35-B3D2-B056CC762AC7

SPRING2014_CSCI570_EXAM2

#228 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

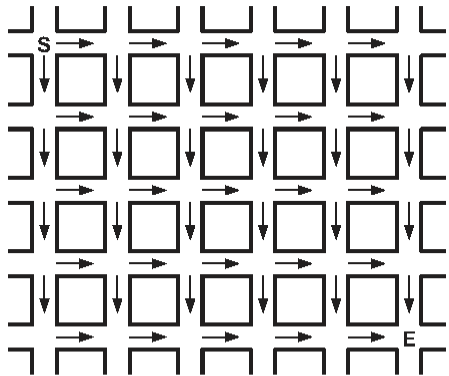


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

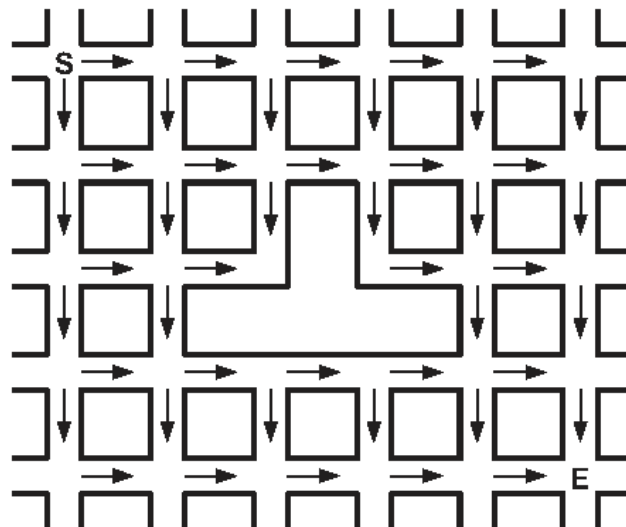


Figure B.



4209CB9C-07C0-4127-B2B9-6DF10464DFA4

SPRING2014_CSCI570_EXAM2

#228 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

942EB117-2651-4B13-8F86-5ECC2CF369F4

SPRING2014_CSCI570_EXAM2

#228

9 of 10



Additional Space



7B3C9EA6-CA52-4D0E-A7CB-54EF394A776B

SPRING2014_CSCI570_EXAM2

#228 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



7C169C6B-702E-4A1F-8B98-4715AC7364D6

SPRING2014_CSCI570_EXAM2

#229

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



1B35F9D5-89C9-4A88-AA23-A0165ECCCB50

SPRING2014_CSCI570_EXAM2

#229

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

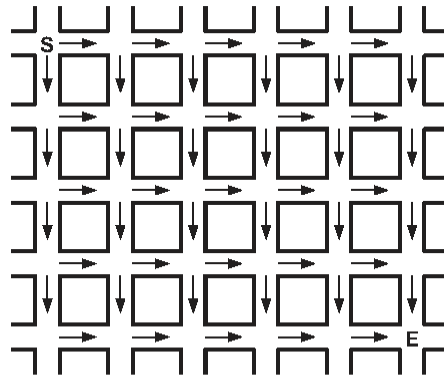


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

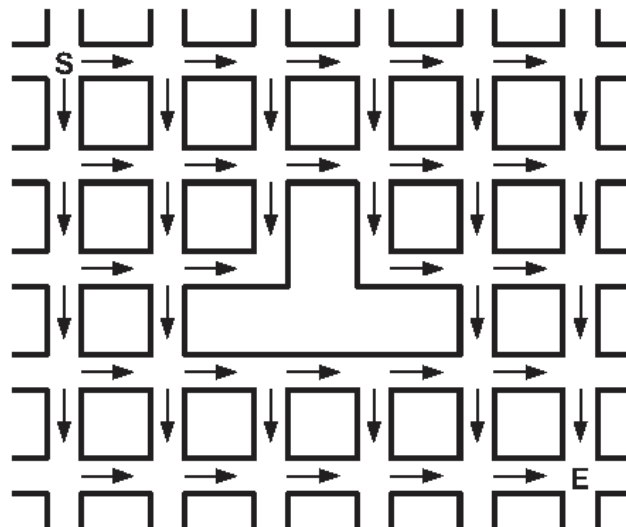


Figure B.



B3C847B3-D636-437C-982B-674CC57AF675

SPRING2014_CSCI570_EXAM2

#229 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

7EA941BB-E0D0-4C6D-BA38-2D71D267A370

SPRING2014_CSCI570_EXAM2

#229

9 of 10



Additional Space



011A4A50-B0B2-405D-920E-4A9903D59262

SPRING2014_CSCI570_EXAM2

#229 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



4017819B-DE2B-400E-9265-E365952018AE

SPRING2014_CSCI570_EXAM2

#230 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



925EA157-4AF1-43FC-99A1-8DDFCF08B439

SPRING2014_CSCI570_EXAM2

#230 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

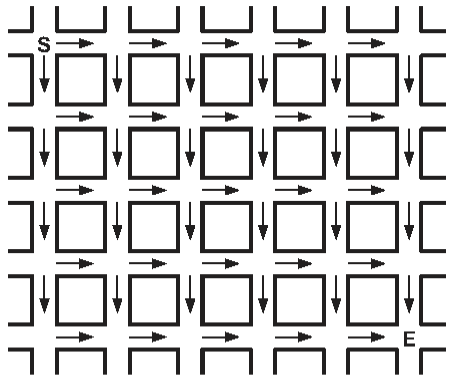


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



E9A4BA6A-663A-449A-8DD2-04E846E01D6E

SPRING2014_CSCI570_EXAM2

#230 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

40A06015-B911-4309-A994-2B3D277623EA

SPRING2014_CSCI570_EXAM2

#230

9 of 10



Additional Space



CB9F03A6-2FBC-41B5-A246-1F66CD932CE0

SPRING2014_CSCI570_EXAM2

#230 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



AEBEE634-713E-4A8A-A06D-C019DD1EF17A

SPRING2014_CSCI570_EXAM2

#231 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



752D6B15-4AD0-44FD-A140-9BCDD916DBB1

SPRING2014_CSCI570_EXAM2

#231 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



F2999846-F6D4-4643-8F4A-312BCBF3ECE6

SPRING2014_CSCI570_EXAM2

#231 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

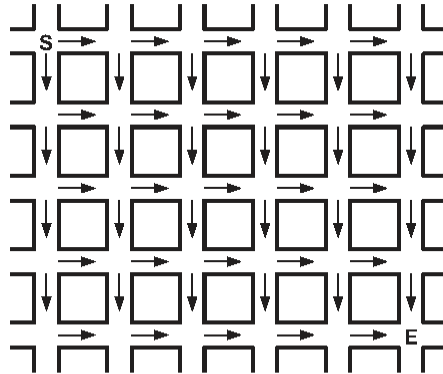


Figure A.

-

Figure B.



3DC0A24D-5444-42A8-8B9B-66DD93DC5362

SPRING2014_CSCI570_EXAM2

#231 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

ECFE7365-D29D-483A-8C86-1AE02C1FA407

SPRING2014_CSCI570_EXAM2

#231

9 of 10



Additional Space



19E06F1F-2235-4F91-9233-03A7390DD671

SPRING2014_CSCI570_EXAM2

#231 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



4C6E22A7-BB04-44DF-B1F7-6B7C06BB58EC

SPRING2014_CSCI570_EXAM2

#232 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



ABED8689-5648-4857-8C5F-0ECBB6F8BD3C

SPRING2014_CSCI570_EXAM2

#232 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

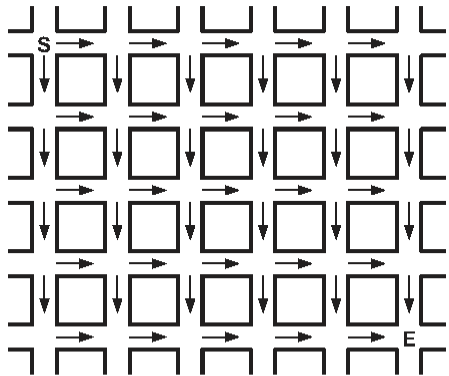


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



02DEC1B5-31EC-4E54-80C7-217E56A1B801

SPRING2014_CSCI570_EXAM2

#232 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

28E0987E-472C-49E6-A757-DF7BCB3CB667

SPRING2014_CSCI570_EXAM2

#232

9 of 10



Additional Space



DAD68603-55BA-4D9C-BA75-40D78C766D51

SPRING2014_CSCI570_EXAM2

#232 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



2202C61C-FCC5-4C68-98ED-A132F8E66853

SPRING2014_CSCI570_EXAM2

#233 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



5) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

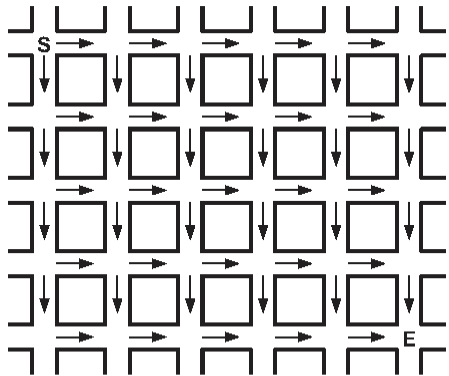


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



DF339772-23CD-4287-9BDF-A7369DA799DA

SPRING2014_CSCI570_EXAM2

#233 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

4235CBB4-3B0D-43F0-9AB5-10B6CD499DB6

SPRING2014_CSCI570_EXAM2

#233

9 of 10



Additional Space



BC3678B2-C31C-4982-9105-96DE4698AAC1

SPRING2014_CSCI570_EXAM2

#233 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



0A3EE1B5-0AED-4D2B-8905-09CEF6F0EFC6

SPRING2014_CSCI570_EXAM2

#234 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



8A280744-EE94-4B2E-BD76-84EA2375BED9

SPRING2014_CSCI570_EXAM2

#234 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

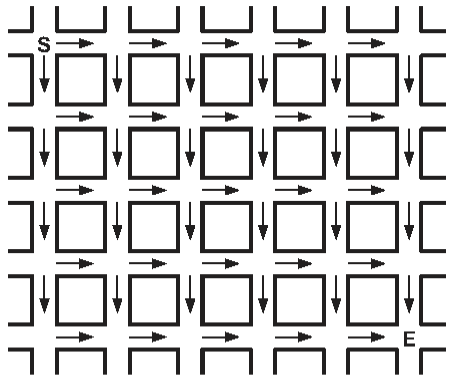


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

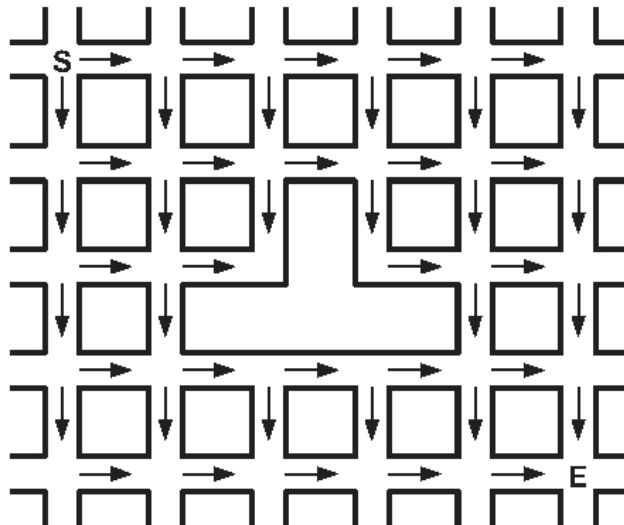


Figure B.



E0F62080-7470-4D09-9DF3-AEA0E91C1AFC

SPRING2014_CSCI570_EXAM2

#234 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

6BC46D83-178C-4643-971E-EA6D2C16D84F

SPRING2014_CSCI570_EXAM2

#234

9 of 10



Additional Space



B08C8800-4F17-4BDA-B3C1-DE0D33A3909F

SPRING2014_CSCI570_EXAM2

#234 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



0ED0FE41-12C2-4A8C-B67C-96A826360EE9

SPRING2014_CSCI570_EXAM2

#235 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



EDE2B21F-496F-4174-A44F-F4ADFE8B2DFD

SPRING2014_CSCI570_EXAM2

#235 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

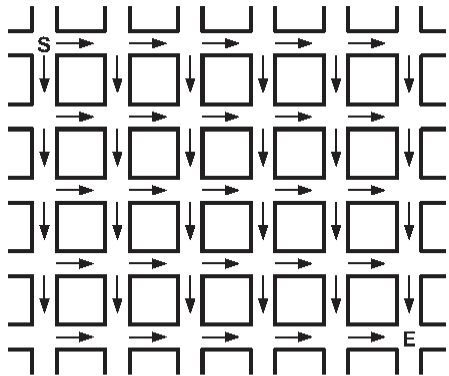


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

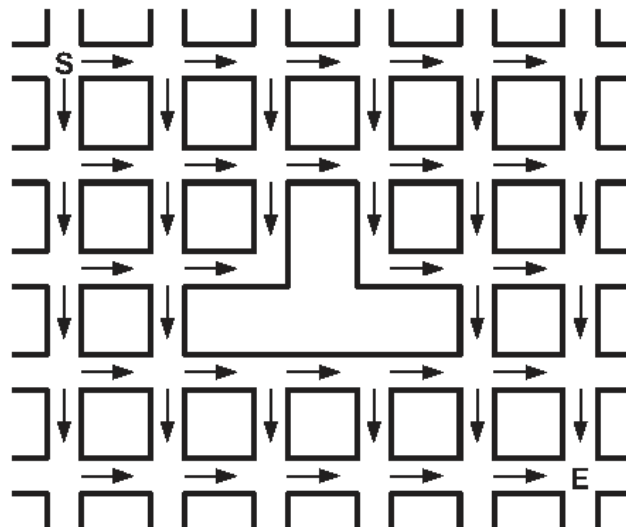


Figure B.



AF550A21-B8E1-436B-9377-C311B29ADCB3

SPRING2014_CSCI570_EXAM2

#235 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

C7F3FD68-D23E-445A-B7E9-40FC7989B8BF

SPRING2014_CSCI570_EXAM2

#235

9 of 10



Additional Space



BBF664D6-CF09-4C66-B759-3DCE3CFA81E7

SPRING2014_CSCI570_EXAM2

#235 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



17374FA0-6C53-4B91-B90A-0686689CDEB9

SPRING2014_CSCI570_EXAM2

#236 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

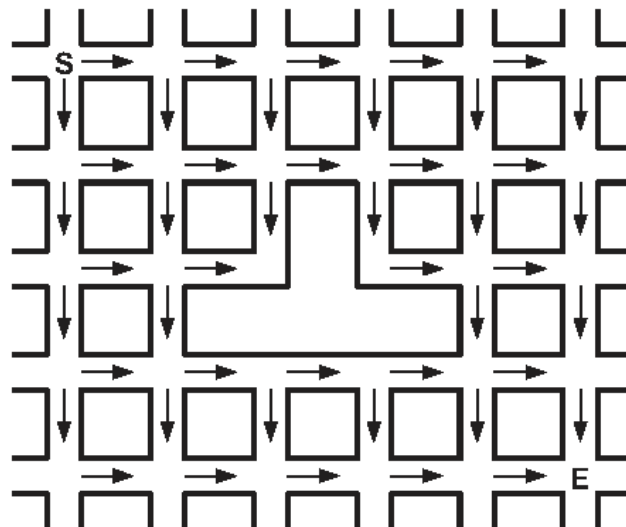


Figure B.



A65F8F44-5C66-4AC6-AE76-AD41476D18AA

SPRING2014_CSCI570_EXAM2

#236 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

31FB6161-E5C1-402B-8F00-120C4132A132

SPRING2014_CSCI570_EXAM2

#236

9 of 10



Additional Space



E5E7CF05-7EF9-4099-87E3-E2DECDF4B96B

SPRING2014_CSCI570_EXAM2

#236 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



D8E1D041-4C2C-414F-A97A-9651F543C6C1

SPRING2014_CSCI570_EXAM2

#237 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



38166D89-1719-4A56-81B4-7A439C9E72AC

SPRING2014_CSCI570_EXAM2

#237 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

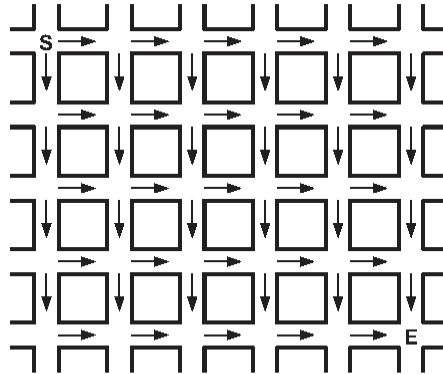


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

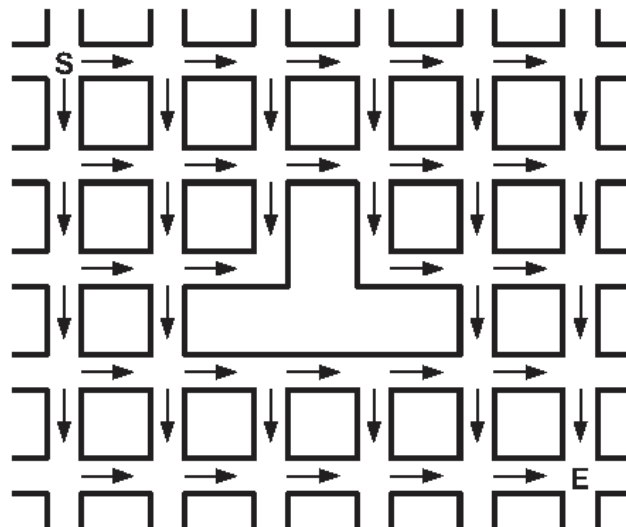


Figure B.



10A30FBD-1EC9-4EDD-B171-4BE69C8D72CD

SPRING2014_CSCI570_EXAM2

#237 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

BD1D1FD7-FB70-4D24-AC07-2CB501E7CC6E

SPRING2014_CSCI570_EXAM2

#237

9 of 10



Additional Space



61E1B63B-ED76-499C-9053-A4BE074FEB29

SPRING2014_CSCI570_EXAM2

#237 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



894F8023-B5A0-474B-A228-68E5ED1B6528

SPRING2014_CSCI570_EXAM2

#238 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



DB7A1C15-963E-44BF-B2BF-CBE9BAA625D3

SPRING2014_CSCI570_EXAM2

#238

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

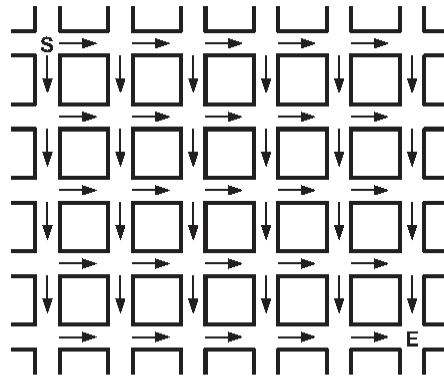


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

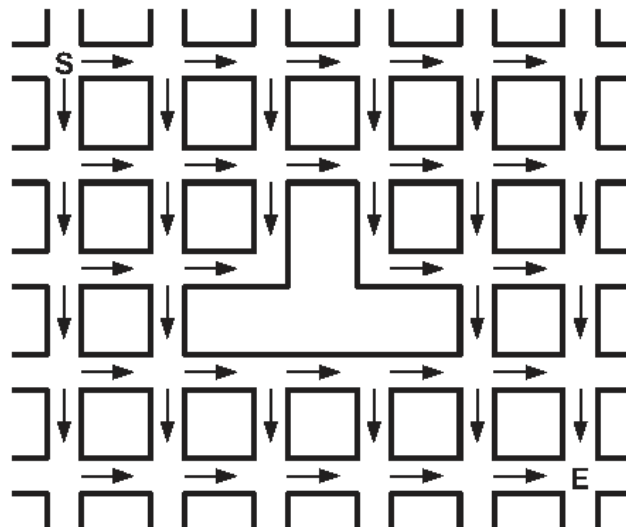


Figure B.



55D7F19A-C25B-4654-B563-AF2FBE84BC7A

SPRING2014_CSCI570_EXAM2

#238 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9F94D72C-4912-4E42-B935-24C4985CA4FD

SPRING2014_CSCI570_EXAM2

#238

9 of 10



Additional Space



872455B2-3781-4712-A7F0-E97BF23A72FA

SPRING2014_CSCI570_EXAM2

#238 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



F8449A41-20FE-44B2-A3F4-E6A7F5C6FD96

SPRING2014_CSCI570_EXAM2

#239 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



6F23710D-BFC0-4355-B455-6ED20F35140F

SPRING2014_CSCI570_EXAM2

#239 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

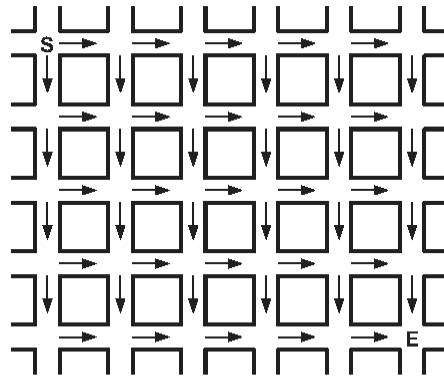


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



FE9154C1-A8B8-425D-BCD6-DF582C85347

SPRING2014_CSCI570_EXAM2

#239 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

4A5AA9A2-F3D0-4882-B2C0-18EAB8B42B19

SPRING2014_CSCI570_EXAM2

#239

9 of 10



Additional Space



5BD195DA-C80C-46E8-B461-3D62F7FA307B

SPRING2014_CSCI570_EXAM2

#239 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



92299C14-6BFE-4F17-AF9B-BA9EA46FEE1C

SPRING2014_CSCI570_EXAM2

#240 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



0FA78155-E32D-4AF6-ABBF-7CE48E59DE95

SPRING2014_CSCI570_EXAM2

#240 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

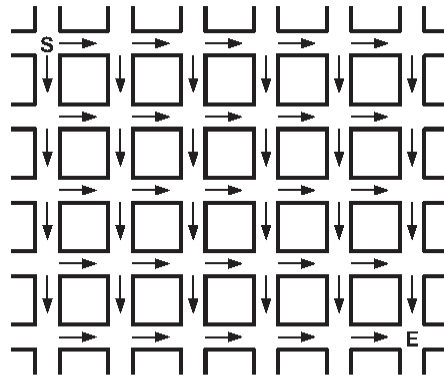


Figure A.



384BB8CB-AC12-4749-B893-9BA890D5A65D

SPRING2014_CSCI570_EXAM2

#240 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

D5B734B6-7D5A-40B4-A691-FA7CC77F0036

SPRING2014_CSCI570_EXAM2

#240

9 of 10



Additional Space



06994B94-96DC-4195-A893-91333EDCE64E

SPRING2014_CSCI570_EXAM2

#240 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



152782C5-BE55-4AD9-A191-F950690C1BE0

SPRING2014_CSCI570_EXAM2

#241 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

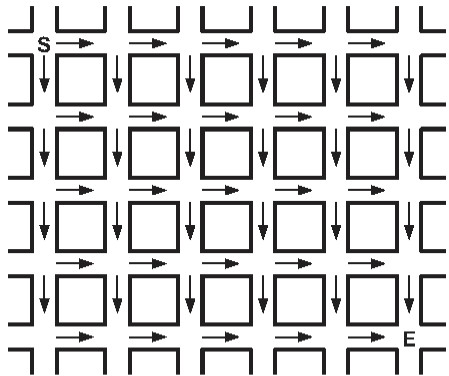


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



44F591FB-42A6-492F-862F-5D76E247B32B

SPRING2014_CSCI570_EXAM2

#241 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

CBC139CB-CC75-4C47-B09A-FAF0F7144140

SPRING2014_CSCI570_EXAM2

#241

9 of 10



Additional Space



A3F0A481-8FA8-4402-B401-6FE3BC6691CC

SPRING2014_CSCI570_EXAM2

#241 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



79632E31-E024-4FC3-8CCE-749CE11F5DFE

SPRING2014_CSCI570_EXAM2

#242 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



BE7038BD-8F94-4922-AEB8-33BAD55F6E61

SPRING2014_CSCI570_EXAM2

#242 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

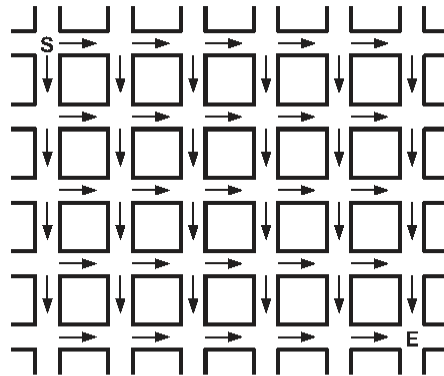


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

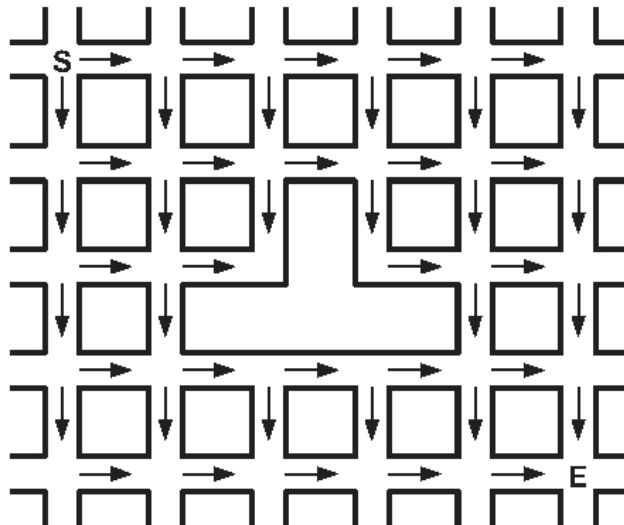


Figure B.



30DD3177-7DEC-47C2-9D87-97BCD03E58F3

SPRING2014_CSCI570_EXAM2

#242 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

727AEF2D-1A4A-4230-96EF-69731C298524

SPRING2014_CSCI570_EXAM2

#242

9 of 10



Additional Space



25D53B98-FFB6-4CE5-A5FB-2A1D2099ECC9

SPRING2014_CSCI570_EXAM2

#242 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



37D83C96-D45B-4C96-898C-73A156EAE53B

SPRING2014_CSCI570_EXAM2

#243 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



B46F76EF-C095-4A93-865F-EDBA564DCB8B

SPRING2014_CSCI570_EXAM2

#243

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

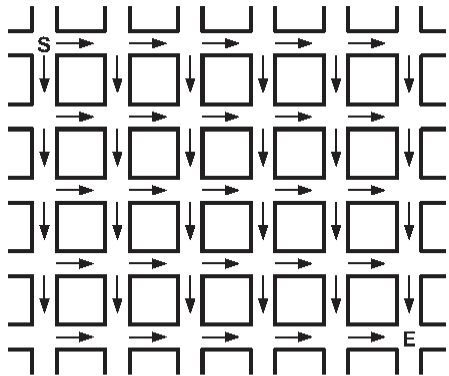


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



2B195E94-7486-4827-B24F-B254D78E60FD

SPRING2014_CSCI570_EXAM2

#243 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

A983EB70-F447-40A2-A97A-310D12C745DE

SPRING2014_CSCI570_EXAM2

#243

9 of 10



Additional Space



5FD02B4F-3844-4667-B90B-6B3303343BEC

SPRING2014_CSCI570_EXAM2

#243 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



2085B7BA-05B7-4974-AD33-1A4726EB3F38

SPRING2014_CSCI570_EXAM2

#244 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)



Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

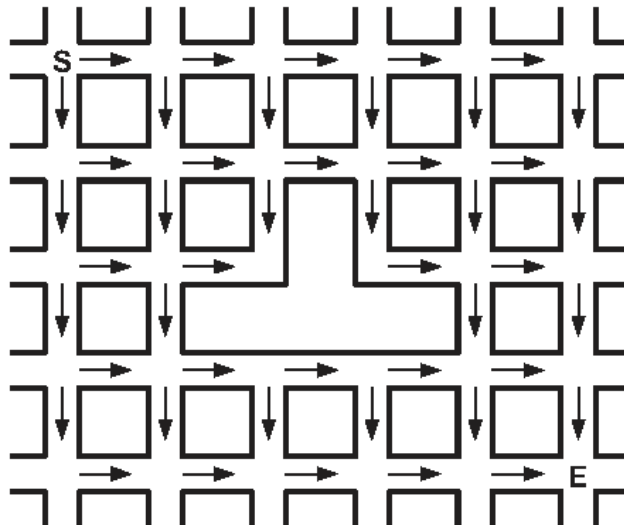


Figure B.



E483AC68-9A0E-4F2C-9B2C-E9AEC3A0ED13

SPRING2014_CSCI570_EXAM2

#244 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

4905AC5F-AA4B-4780-87A1-F185CD076D15

SPRING2014_CSCI570_EXAM2

#244

9 of 10



Additional Space



2784D5EE-DAE7-4BA9-9A96-FDD794B48F5D

SPRING2014_CSCI570_EXAM2

#244 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



8E9DC5C9-A7C7-4ED7-BCBA-D911757B38A7

SPRING2014_CSCI570_EXAM2

#245

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



80D0C6C1-FF37-42F6-BAB6-CE98A9D28F9E

SPRING2014_CSCI570_EXAM2

#245 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

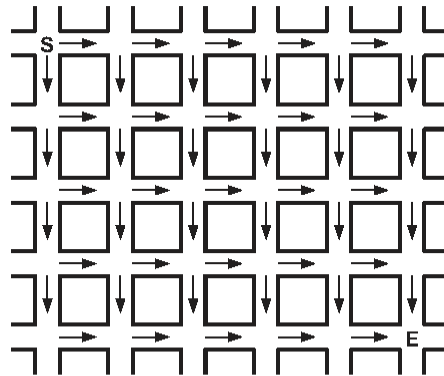


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

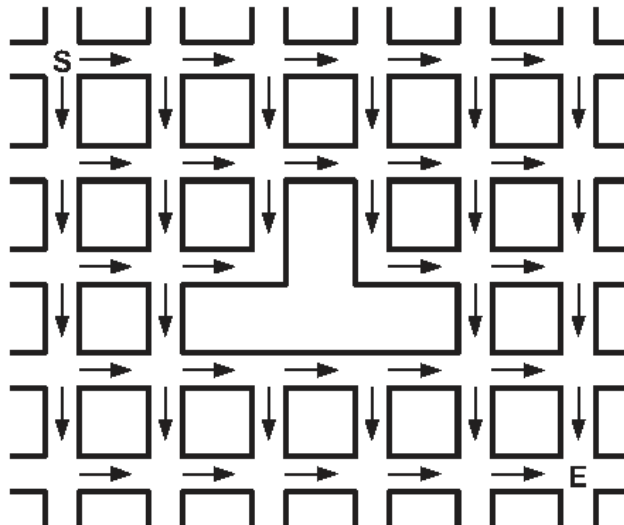


Figure B.



B7A8D992-FC3C-4235-A7AC-F07D674D0B33

SPRING2014_CSCI570_EXAM2

#245 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

95E01DC2-C525-46AF-A16B-C8632A7A9589

SPRING2014_CSCI570_EXAM2

#245

9 of 10



Additional Space



BC414AB2-AD52-4E64-B527-720E4D2BDA9A

SPRING2014_CSCI570_EXAM2

#245 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



E1817DC3-FF16-451A-B465-D8095EC8535C

SPRING2014_CSCI570_EXAM2

#246 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



4C62A5A1-E472-4D6A-8B9C-98B9690D6EC3

SPRING2014_CSCI570_EXAM2

#246

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

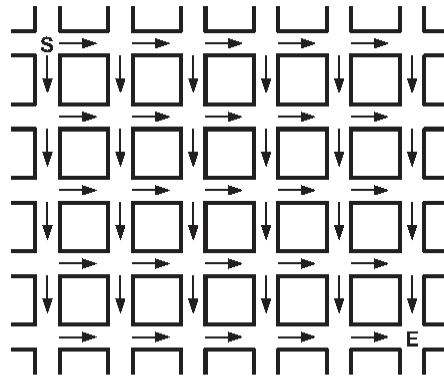


Figure A.



9BEC2DF-77AB-4AE2-80F6-D19C9CC5E0FB

SPRING2014_CSCI570_EXAM2

#246 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

67454980-A822-4AD1-AD35-56D0A95AF73F

SPRING2014_CSCI570_EXAM2

#246

9 of 10



Additional Space



662A17B9-AC9C-4925-A562-E0B02995D6A4

SPRING2014_CSCI570_EXAM2

#246 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



6EB717C5-96C8-40CA-9624-B849F46E8BAF

SPRING2014_CSCI570_EXAM2

#247 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



10D9A86A-3D86-4AE2-B06D-5115CB3D3E72

SPRING2014_CSCI570_EXAM2

#247 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



6A9792A8-1ECB-4383-ACB4-91D2C7904FEC

SPRING2014_CSCI570_EXAM2

#247

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

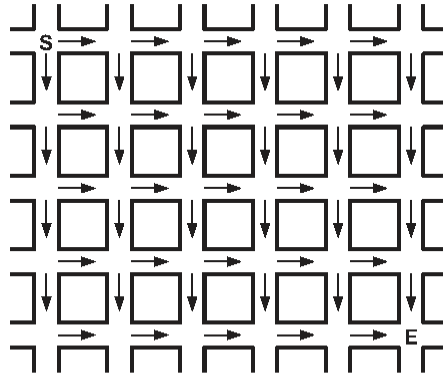


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\cdot, \cdot)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\cdot, \cdot)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



533F2CE2-DE03-4F9A-BB4B-03F2806EDFC2

SPRING2014_CSCI570_EXAM2

#247 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

9033569B-30AB-4328-9621-B2BB806D08E0

SPRING2014_CSCI570_EXAM2

#247

9 of 10



Additional Space



9E2BB45C-4917-4589-BBA0-9CDF6BEB213

SPRING2014_CSCI570_EXAM2

#247 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



CB9D58A0-221A-4388-BB21-64EA1602E94F

SPRING2014_CSCI570_EXAM2

#248

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

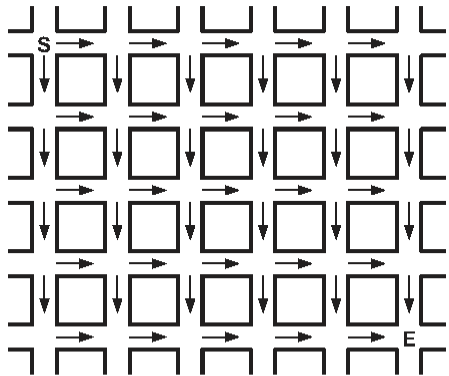


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



E9292228-49A4-4EFE-9524-17331E64065C

SPRING2014_CSCI570_EXAM2

#248 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

5E8769AA-31F7-44CE-ADE1-A09F2378E4D6

SPRING2014_CSCI570_EXAM2

#248

9 of 10



Additional Space



8D6C7203-B63D-4062-80B7-16AA98A61AEC

SPRING2014_CSCI570_EXAM2

#248 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



82BDA7FC-9D3E-4FF9-A142-49763CD21C83

SPRING2014_CSCI570_EXAM2

#249

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



E5A805F9-DEA7-4CE1-AE29-F8718F73538E

SPRING2014_CSCI570_EXAM2

#249

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

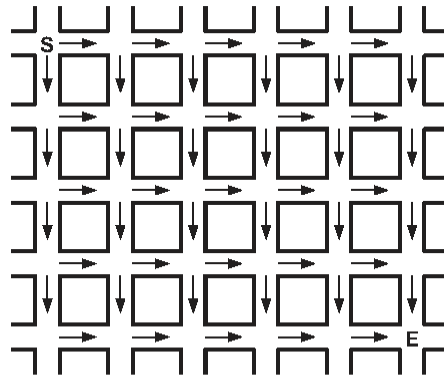


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



45B856BD-8C7F-4D94-B2F0-D77E5D9880A3

SPRING2014_CSCI570_EXAM2

#249 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

82EBCD6A-87A9-4877-BD09-EC0DEF01A32

SPRING2014_CSCI570_EXAM2

#249

9 of 10



Additional Space



281F66FF-7B93-4110-AF56-6D82B0CE9345

SPRING2014_CSCI570_EXAM2

#249 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



6704C113-72B0-45CD-ABA2-728C0C4DD2C8

SPRING2014_CSCI570_EXAM2

#250

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



BB981AB7-D290-4C01-8A6F-D70D2F058C38

SPRING2014_CSCI570_EXAM2

#250

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

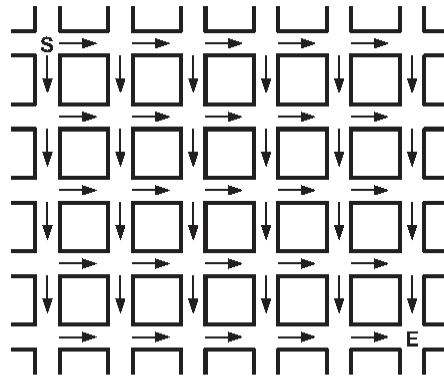


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



1A64A25B-9102-4989-927B-EC16A64F5772

SPRING2014_CSCI570_EXAM2

#250 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

28EC4814-F22F-45AA-8250-6957BD449818

SPRING2014_CSCI570_EXAM2

#250

9 of 10



Additional Space



C7D0A350-F004-434B-A85A-4F4D40AF9FD3

SPRING2014_CSCI570_EXAM2

#250 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



79B813BE-0E8C-4419-914B-98FDC0B95646

SPRING2014_CSCI570_EXAM2

#251 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



7C595E82-645B-4A0A-8BEF-BB896DB50D93

SPRING2014_CSCI570_EXAM2

#251 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

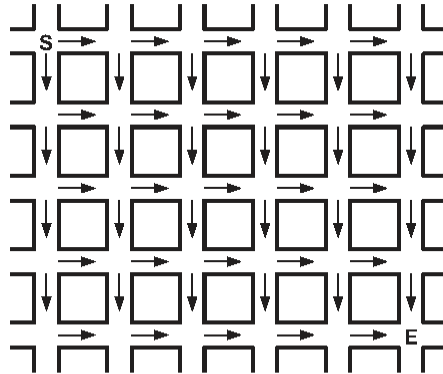


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



F8DD0323-9918-4061-9914-958F04B0286D

SPRING2014_CSCI570_EXAM2

#251 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

0ECE66A7-AF23-46AB-B6F4-F7371655010B

SPRING2014_CSCI570_EXAM2

#251

9 of 10



Additional Space



069E9978-DAAE-4C3B-8D56-37C0C1F58004

SPRING2014_CSCI570_EXAM2

#251 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



B638FF64-E83B-4C6D-8B20-5BDD5B15A036

SPRING2014_CSCI570_EXAM2

#252 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

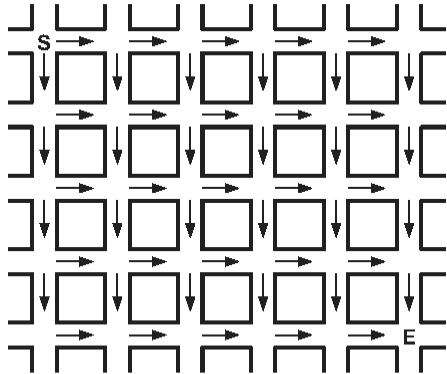


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



A4F850AC-E68B-4D7B-ABFE-E78696C9FF17

SPRING2014_CSCI570_EXAM2

#252 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

4696A4CF-9090-4663-9FF2-02BDABBC2A15

SPRING2014_CSCI570_EXAM2

#252

9 of 10



Additional Space



C6DF5619-0D97-4A6C-8122-F51EBAFF2805

SPRING2014_CSCI570_EXAM2

#252 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



22F32328-CE45-4A8B-90AE-0B23618F6313

SPRING2014_CSCI570_EXAM2

#253 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

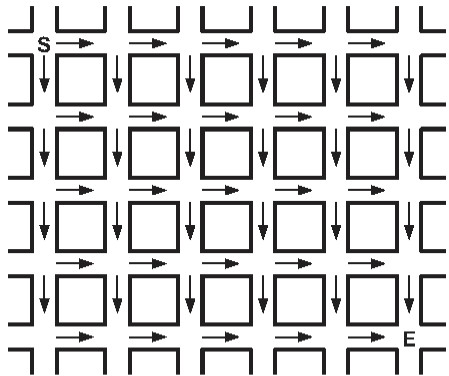


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

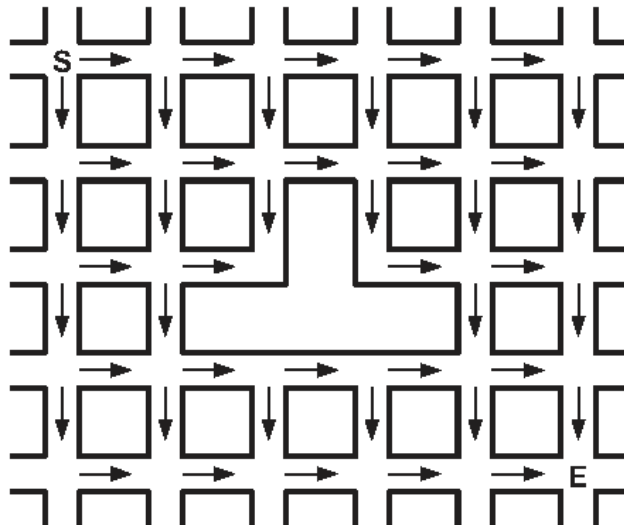


Figure B.



54B8B7C5-FEF2-4662-B824-5547EA8E4DF8

SPRING2014_CSCI570_EXAM2

#253 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

30F27AD2-14A9-4513-B8B8-48031D028687

SPRING2014_CSCI570_EXAM2

#253

9 of 10



Additional Space



F332F596-083A-429E-A3C7-BB63A7C1F826

SPRING2014_CSCI570_EXAM2

#253 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



49A05AF7-CA7C-4307-8915-ACFA7971D5E1

SPRING2014_CSCI570_EXAM2

#254

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

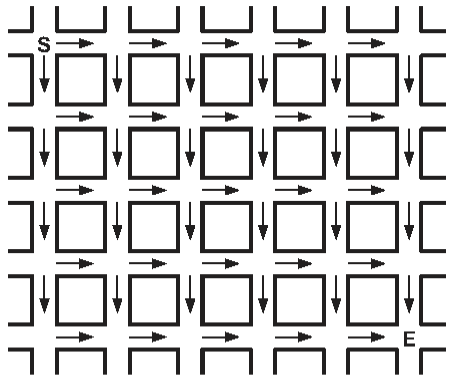


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

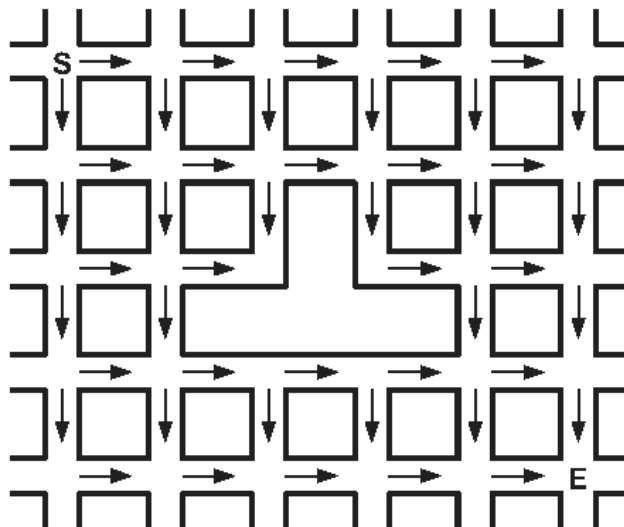


Figure B.



1C252ED2-F8B1-415E-9A82-133A435543B2

SPRING2014_CSCI570_EXAM2

#254 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

3E078FB4-9F8F-49E3-B1D4-94B372239B0B

SPRING2014_CSCI570_EXAM2

#254

9 of 10



Additional Space



FDB11DFF-447B-4DC7-AB66-22393F64F138

SPRING2014_CSCI570_EXAM2

#254 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



EC5A235F-028C-4AF9-884E-36368CDE47B2

SPRING2014_CSCI570_EXAM2

#255 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



289D576C-02E9-4EDB-8D58-89228047A2C8

SPRING2014_CSCI570_EXAM2

#255 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

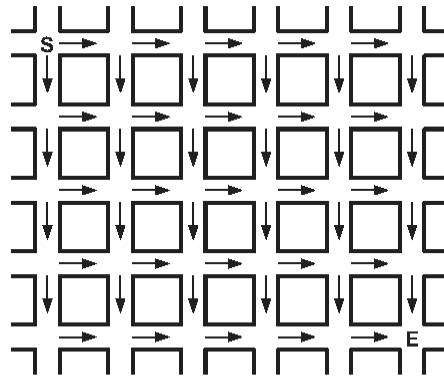


Figure A.



405A5BBE-CC51-4D85-9684-5244FE9A1B2D

SPRING2014_CSCI570_EXAM2

#255 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

2F866BF9-559E-479D-AF07-EA4EC6A802EF

SPRING2014_CSCI570_EXAM2

#255

9 of 10



Additional Space



0649E8FA-352D-4BD5-BB3C-D22C8542CAF1

SPRING2014_CSCI570_EXAM2

#255 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



963AE2C6-A017-4000-8D29-08487B07F262

SPRING2014_CSCI570_EXAM2

#256

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



463A4399-4708-481D-BA64-7E8E2FCF1F53

SPRING2014_CSCI570_EXAM2

#256 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

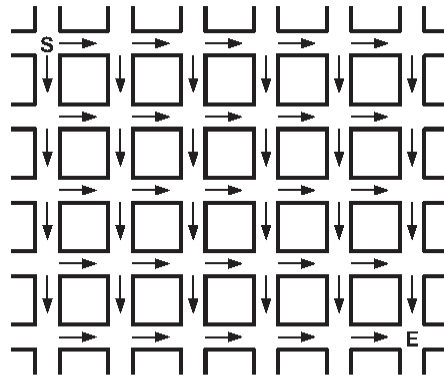


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



02E7B3DB-8880-4560-9CD1-B219C2EEDD98

SPRING2014_CSCI570_EXAM2

#256 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

D40CEDB6-773C-4E37-880D-3434BE09A954

SPRING2014_CSCI570_EXAM2

#256

9 of 10



Additional Space



B74173CA-8E8B-4D33-8938-FCC3E230712A

SPRING2014_CSCI570_EXAM2

#256 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



E4CD5E3C-A33B-4BEE-A3F8-3657167DF2BC

SPRING2014_CSCI570_EXAM2

#257 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



CE2826E2-302E-4E2C-9D20-B4FE1CEF1D0F

SPRING2014_CSCI570_EXAM2

#257 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

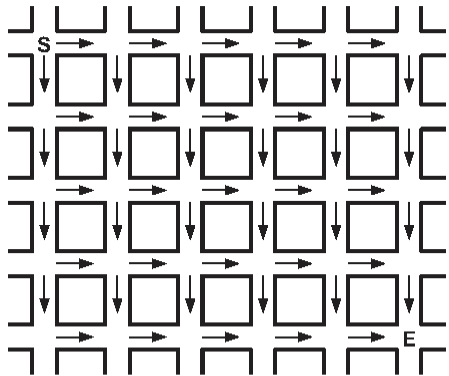


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



12FCFDE7-968F-4463-A210-BEDD9A0C4B3E

SPRING2014_CSCI570_EXAM2

#257 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

DBA3ACEC-DC05-4769-82C5-E3B54138A166

SPRING2014_CSCI570_EXAM2

#257

9 of 10



Additional Space



06A5F70B-C4C9-42ED-915B-822368C53A68

SPRING2014_CSCI570_EXAM2

#257 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



D53FD520-9A2F-4A16-A1FE-8A1605DE321E

SPRING2014_CSCI570_EXAM2

#258

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



ED7BD24A-3286-4138-AF35-80A479AF4951

SPRING2014_CSCI570_EXAM2

#258

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

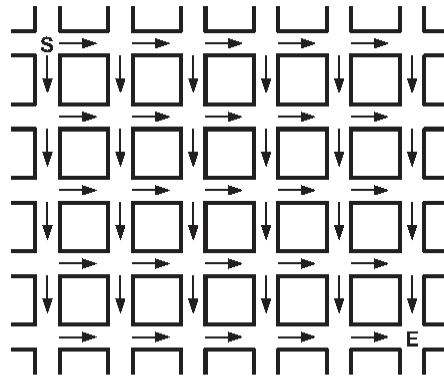


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

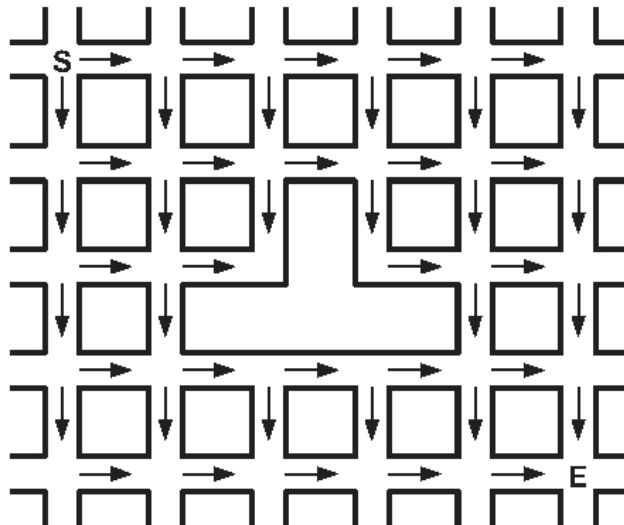


Figure B.



E7ECE212-2B7A-4EE6-8A67-D04343EAB298

SPRING2014_CSCI570_EXAM2

#258 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

16CBD9B0-8F9C-40A8-AE16-885F10DC3452

SPRING2014_CSCI570_EXAM2

#258

9 of 10



Additional Space



79AB9351-1348-45E1-8A54-2DA50C68F433

SPRING2014_CSCI570_EXAM2

#258 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



2C70DC07-9B3C-40AF-B7CD-118837EDBDB9

SPRING2014_CSCI570_EXAM2

#259

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

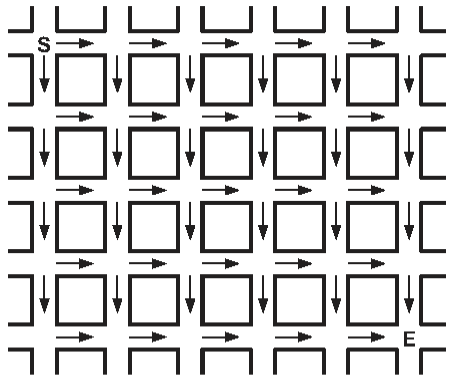


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



BADB9C09-225D-44F6-9ACC-D0F4D59F5B41

SPRING2014_CSCI570_EXAM2

#259 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

72B59BEB-B106-4C14-87FB-2F224AA4C043

SPRING2014_CSCI570_EXAM2

#259

9 of 10



Additional Space



BA1B2E82-3BAD-4AB5-BAC7-977A538FC6D2

SPRING2014_CSCI570_EXAM2

#259 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



A4A72A45-A42E-4E4A-9B2D-BEB4CAA5EAD8

SPRING2014_CSCI570_EXAM2

#260 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



B1316C1C-FB68-494B-84CC-1B9DDDF235F1

SPRING2014_CSCI570_EXAM2

#260 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

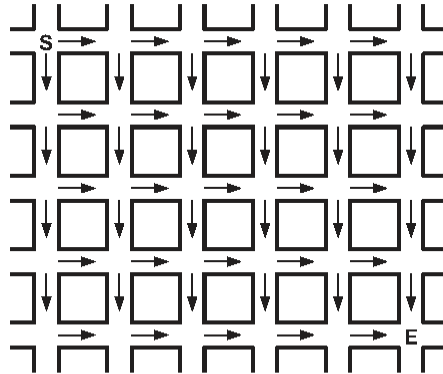


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)
- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)

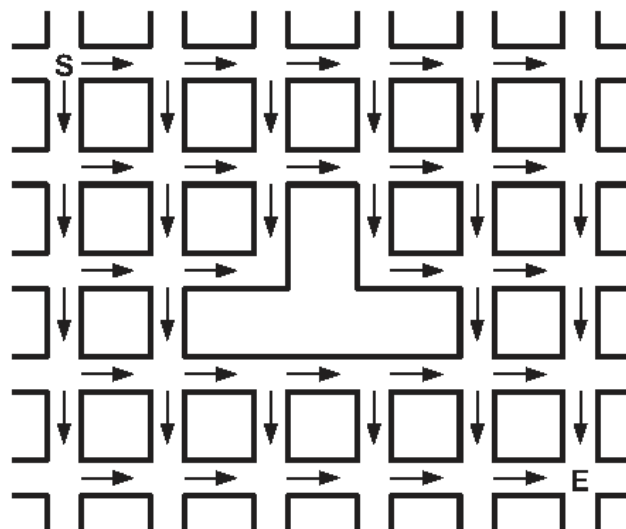


Figure B.



68032C46-3019-4119-93C9-BAC8607A3B05

SPRING2014_CSCI570_EXAM2

#260 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

0F7FED18-9378-419E-BFE7-754B0831E4BB

SPRING2014_CSCI570_EXAM2

#260

9 of 10



Additional Space



BDE7C73F-4C46-4C1B-9CBE-0F4139A194C8

SPRING2014_CSCI570_EXAM2

#260 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



D2BB0693-B511-4374-8E08-BAB290F96AAC

SPRING2014_CSCI570_EXAM2

#261 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



13E01191-E964-49A0-9A73-D725DD65C508

SPRING2014_CSCI570_EXAM2

#261 6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

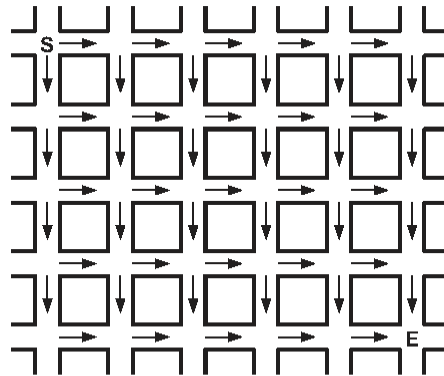


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



4CF99E01-B1C0-498C-B891-4434FF1C35C9

SPRING2014_CSCI570_EXAM2

#261 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

DE40A6EE-569B-4DC4-B343-D176A25805F4

SPRING2014_CSCI570_EXAM2

#261

9 of 10



Additional Space



AC28D24A-ED6B-49DE-9081-557970D6519A

SPRING2014_CSCI570_EXAM2

#261 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



C98CAD6D-EE67-4A4C-8AC7-1F547335BB10

SPRING2014_CSCI570_EXAM2

#262 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

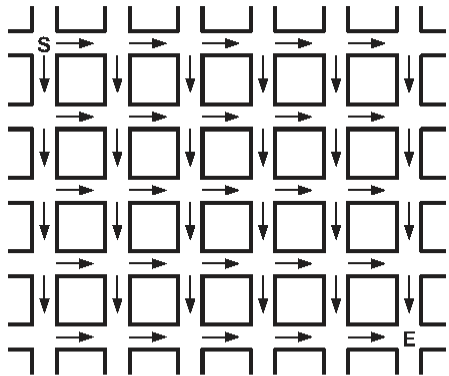


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



FD0B2A2F-11CC-4585-9461-3F028137F595

SPRING2014_CSCI570_EXAM2

#262 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

BFA2D060-81CA-41D6-AEDD-D89308BDF837

SPRING2014_CSCI570_EXAM2

#262

9 of 10



Additional Space



3E294C3B-6DA7-42D2-AA98-762D9E0BB9FD

SPRING2014_CSCI570_EXAM2

#262 10 of 10



CS570
Analysis of Algorithms
Spring 2014
Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



5B7A9D8C-C577-4598-A0AF-31F11A17D071

SPRING2014_CSCI570_EXAM2

#263 2 of 10

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



17321513-ABC5-47DA-9FD5-3ACE3E102D99

SPRING2014_CSCI570_EXAM2

#263 4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



- c) 15 pts
- You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.
- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

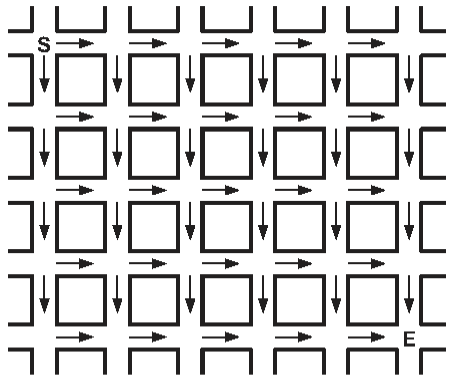


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



2569AEC7-9980-49A6-A513-5D6D79F5EED2

SPRING2014_CSCI570_EXAM2

#263 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

09AB1403-15A3-436B-BA15-8289703839A2

SPRING2014_CSCI570_EXAM2

#263

9 of 10



Additional Space



721D7BBA-316D-4BB4-B142-0E47FB749904

SPRING2014_CSCI570_EXAM2

#263 10 of 10



CS570

Analysis of Algorithms

Spring 2014

Exam II

Name: _____

Student ID: _____

____ On Campus ____ DEN

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	20	
Problem 5	15	
Problem 6	15	
Total	100	

2 hr exam

Closed book and notes

If a description to an algorithm is required please limit your description to within 150 words, anything beyond 150 words will not be considered.



1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[**TRUE/FALSE**]

Suppose $f(n) = f\left(\frac{n}{2}\right) + 56$, then $f(n) = \Theta(n)$

[**TRUE/FALSE**]

Maximum value of an s-t flow could be less than the capacity of a given s-t cut in a flow network.

[**TRUE/FALSE**]

For any edge e that is part of the minimum cut in G , if we increase the capacity of that edge by any integer $k > 1$, then that edge will no longer be part of the minimum cut.

[**TRUE/FALSE**]

Any problem that can be solved using dynamic programming has a polynomial worst case time complexity with respect to its input size.

[**TRUE/FALSE**]

In a dynamic programming solution, the space requirement is always at least as big as the number of unique sub problems

[**TRUE/FALSE**]

In the divide and conquer algorithm to compute the closest pair among a given set of points on the plane, if the sorted order of the points on both X and Y axis are given as an added input, then the running time of the algorithm improves to $O(n)$.

[**TRUE/FALSE**]

If f is a max s-t flow of a flow network G with source s and sink t , then the value of the min s-t cut in the residual graph G_f is 0.

[**TRUE/FALSE**]

Bellman-Ford algorithm can solve the shortest path problem in graphs with negative cost edges in polynomial time.

[**TRUE/FALSE**]

Given a directed graph $G=(V,E)$ and the edge costs, if every edge has a cost of either 1 or -1 then we can determine if it has a negative cost cycle in $O(|V|^3)$ time..

[**TRUE/FALSE**]

The space efficient version of the solution to the sequence alignment problem (discussed in class), was a divide and conquer based solution where the divide step was performed using dynamic programming.



2) 15 pts

Given an unlimited supply of coins of denominations x_1, x_2, \dots, x_n , we wish to make change for a value v using at most k coins; that is, we wish to find a set of k coins whose total value is v . This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65. Give an efficient algorithm for the following problem.

Input: $x_1, x_2, \dots, x_n, k, v$.

Question: Is it possible to make change for v using at most k coins, of denominations x_1, x_2, \dots, x_n ? (No proof of correctness required)



928FC7C7-765D-4DB5-B319-C7B6C683855D

SPRING2014_CSCI570_EXAM2

#264

4 of 10

3) 15 pts

Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night. Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people $\{p_1, p_2, \dots, p_n\}$ and the nights $\{d_1, d_2, \dots, d_n\}$. Then for each person p_i , there is a set of nights $S_i \subseteq \{d_1, d_2, \dots, d_n\}$ where p_i is unable to cook. A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_i$.

Describe an algorithm to determine if there is a feasible dinner schedule or not. What is the running time of your procedure?



4) 20 pts

The input to your algorithm is an array $A[1..n]$ of n numbers. You are further assured that $n > 2$, $A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. A local minimum of the array A is defined as an index i such that $1 < i < n$, $A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

a) Prove that a local minimum of A always exists. (5 pts)

b) Design an algorithm that computes a local minimum while making at most $O(\log n)$ pairwise comparisons between elements of A . (no proof required) (15 pts)



5FEF950D-AECC-40F1-9B3E-0EC4EF961E09

SPRING2014_CSCI570_EXAM2

#264

6 of 10

c) 15 pts

You are in Downtown of a city and all the streets are one-way streets. You can only go east (right) on the east-west (left-right) streets, and you can only go south (down) on the north-south (up-down) streets.

- a) In **Figure A** below, how many **unique ways** are there to go from the intersection marked **S** (coordinate (0,0)) to the intersection marked **E** (coordinate (5,4))? (*Hint: You may want to do part (b) first.*) (5 pts)

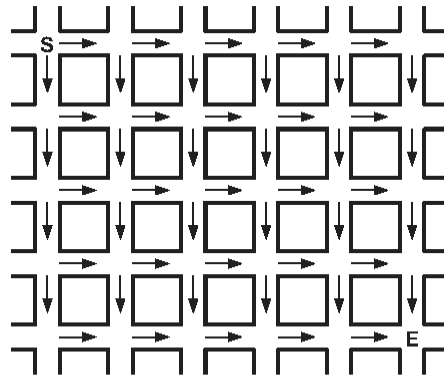


Figure A.



- b) Continuing from part (a), formulate the solution to this problem as a dynamic programming problem, i.e., you need to define what $\text{OPT}(\dots)$ is and write down the **recurrence equation** that expresses $\text{OPT}(\dots)$ in terms of an **optimal** solution to “smaller” subproblems. Please make sure that you include all the boundary conditions and clearly define your notations you use. (*Hint: this is not an optimization problem.*) (5 pts)

- c) Continuing from part (a), in **Figure B** above, how many **unique ways** are there to go from the intersection marked **S** to the intersection marked **E**? You can solve the problem numerically and directly on the figure (*Hint: beware of dead ends.*) (5 pts)



Figure B.



0631BD9D-87A9-46A2-9168-04A04FB97AD7

SPRING2014_CSCI570_EXAM2

#264 8 of 10

b) 15 pts

A spammer is located at one node q in an undirected communication network G and email users are located at a given set of nodes denoted by S . Let c_{uv} denote the effort required to install a spam filter for the network edge (u, v) . The problem is to determine the minimal effort required to isolate the spammer from the email users using the spam filters. Give a polynomial-time algorithm to solve this problem. Justify your algorithm.

A094384D-CBCB-4022-B365-F8A3F9688652

SPRING2014_CSCI570_EXAM2

#264

9 of 10



Additional Space



4110F3C6-7C8B-4FCE-8638-FD9EFF64A315

SPRING2014_CSCI570_EXAM2

#264 10 of 10