CS570 Fall 2021: Analysis of Algorithms Exam II

	Points		Points
Problem 1	20	Problem 4	20
Problem 2	20	Problem 5	20
Problem 3	20		
	Total	**100**	

Instructions:

1. This is a 2-hr exam. Open book and notes and internet access. But no internet communications through social media, chat, or any other form is allowed.
2. If a description to an algorithm or a proof is required, please limit your description or proof to within 150 words, preferably not exceeding the space allotted for that question.
3. No space other than the pages in the exam booklet will be scanned for grading.
4. If you require an additional page for a question, you can use the extra page provided within this booklet. However please indicate clearly that you are continuing the solution on the additional page.
5. Do not detach any sheets from the booklet. Detached sheets will not be scanned.
6. If using a pencil to write the answers, make sure you apply enough pressure, so your answers are readable in the scanned copy of your exam.
7. Do not write your answers in cursive scripts.
8. This exam is printed double sided. Check and use the back of each page.

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

[TRUE/FALSE] (2 pts)
If all edge capacities in a Flow Network are integer multiples of 5 then considering any max flow F in this network, flow over each edge due to F must be an integer multiple of 5.

[TRUE/FALSE] (2 pts)
In a Flow Network G, if the capacity of any edge on a min cut is increased by 1 unit, then the value of max flow in that network will also increase by 1 unit.
Note: an edge on the min cut (A, B) refers to an edge that carries flow out of A (where the source S is) and into B (where the sink T is).

[TRUE/FALSE] (2 pts)
In a Flow Network with maximum flow value v(F)>0, and with more than one min cut, decreasing the capacity of an edge on any of the min cuts will reduce the value of max flow.
Note: an edge on the min cut (A, B) refers to an edge that carries flow out of A (where the source S is) and into B (where the sink T is).

[TRUE/FALSE] (2 pts)
Consider the Ford Fulkerson algorithm and the residual graph used at each iteration. If the Flow Network has no cycles (i.e. the network is a directed acyclic graph), we will not need to include backward edges in the residual graph to achieve max flow.

[TRUE/FALSE] (2 pts)
Suppose an edge e is not saturated due to max flow f in a Flow Network. Then increasing e's capacity will not increase the max flow value of the network.

[TRUE/FALSE] (2 pts)
The time complexity of any dynamic programming algorithm with n2 unique subproblems is $\Omega(n2)$

[TRUE/FALSE] (2 pts)
The Bellman-ford algorithm may not be able to find the shortest simple path in a graph with negative cost edges.

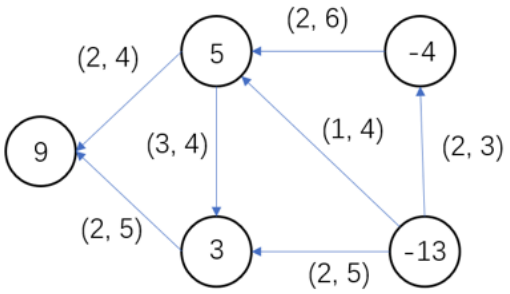
[TRUE/FALSE] (2 pts)
If the running time of an algorithm can be represented as a polynomial in terms of the number of bits in the input, then the algorithm is considered to be efficient.

[TRUE/FALSE] (4 pts)
Recall the coin change problem from lecture where we are trying to pay amount m using the minimum number of coins. We presented two attempts to solve this problem in class, one

based on the greedy approach and one using dynamic programming. If coin denominations are limited to 1, 3, and 4 these two approaches will result in the same number of coins to pay any amount m.

2) 20 pts

In the network G below, the demand values are shown on vertices (supply value if negative). Lower bounds on flow and edge capacities are shown as (lower bound, capacity) for each edge. Determine if there is a feasible circulation in this graph. You need to show all your steps.



a. Reduce the Feasible Circulation with Lower Bounds problem to a Feasible Circulation problem without lower bounds. (8 pts)

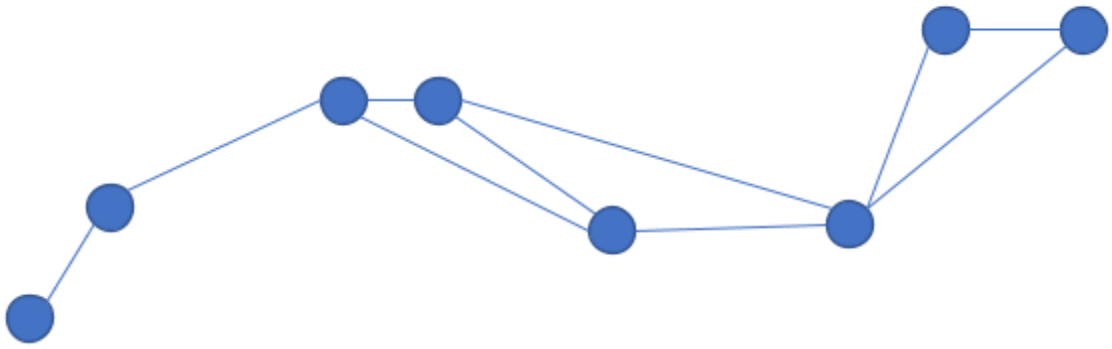
See next page for parts b and c

b. Reduce the Feasible Circulation problem obtained in *part a* to a Maximum Flow problem in a Flow Network. (8 pts)

c. Using the solution to the resulting Max Flow problem explain whether there is a Feasible Circulation in G. (4 pts)

3) 20 pts

In the last 570 exam, there were nearly a thousand students that were ready to take the exam in person at 8 different test locations (L1..L8). There were Si students assigned to each room i. The copy center made some big mistakes and instead of delivering Si papers they delivered Ri papers to each room i, where Ri was higher than Si for some rooms and lower for some others. TAs then had to rush and come up with a solution to redistribute the excess papers at each test location to those test locations that had a shortage. And because the time was short, they ruled out sending papers directly between test locations that were far apart from each other on campus. But they did not rule out sending papers indirectly through other test locations. (For example, L1 and L3 in below graph were considered far apart from each other but papers could still be redistributed between them through L2). So with these considerations in mind, they ended up with a connected graph that looked like this:



a) Assuming that the total number of papers delivered was at least equal to the total number of papers required, design a network flow based algorithm to determine how many papers had to be sent (and in which direction) on each edge in the above network in order to supply each room with the required papers. You need to describe exactly how you reduce this problem to a network flow problem. (14 pts)

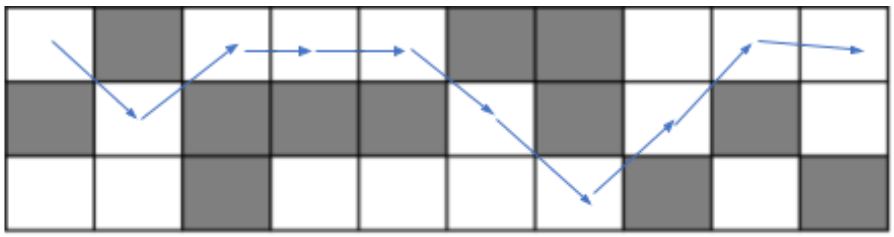
More space provided on next page

Problem 3 continued…

b) Prove that you solution is correct (6 pts)

4) 20 pts

Jack has gotten himself involved in a very dangerous game called the octopus game where he needs to pass a bridge which has some unreliable sections. The bridge consists of 3n tiles as shown below. Some tiles are strong and can withstand Jack's weight, but some tiles are weak and will break if Jack lands on them. Jack has no clue which tiles are strong or weak but we have been given that information in an array called BadTile(3,n) where BadTile (j, i) = 1 if the tile is weak and 0 if the tile is strong. At any step Jack can move either to the tile right in front of him (i.e. from tile (j, i) to (j, i+1)), or diagonally to the left or right (if they exist). (No sideways or backward moves are allowed and one cannot go from tile (1,i) to (3, i+1) or from (3, i) to (1, i+1)). Using dynamic programming find out how many ways (if any) there are for Jack to pass this deadly bridge. Figure below shows bad tiles in gray and one of the possible ways for Jack to safely cross the bridge alive.



a) Define (in plain English) subproblems to be solved. (4 pts)

b) Write a recurrence relation for the subproblems (6 pts)

See next page for parts c and d

c) Using the recurrence formula in part b, write pseudocode using iteration to compute the total number of ways to safely cross the bridge. (5 pts)
Make sure you specify
- base cases and their values (2 pts)
- where the final answer can be found (e.g. opt(n), or opt(0,n), etc.) (1 pt)

d) What is the complexity of your solution? (1 pt)
Is this an efficient solution? (1 pt)

5) 20 pts

Assume a truck with capacity W is loading. There are n packages with different weights, i.e. [w1, w2, … wn], and all the weights are integers. The company's rule requires that the truck needs to take packages with exactly weight W to maximize profit, but the workers like to save their energies for after work activities and want to load as few packages as possible. Assuming that there are combinations of packages that add up to weight W, design an algorithm to find out the minimum number of packages the workers need to load.

a) Define (in plain English) subproblems to be solved. (4 pts)

b) Write a recurrence relation for the subproblems (6 pts)

See next page for parts c and d

c) Using the recurrence formula in part b, write pseudocode using iteration to compute the minimum number of packages to meet the objective. (5 pts)
Make sure you specify
- base cases and their values (2 pts)
- where the final answer can be found (e.g. opt(n), or opt(0,n), etc.) (1 pt)

d) What is the complexity of your solution? (1 pt)
Is this an efficient solution? (1 pt)

Additional Space

Additional Space

Additional Space