

CS570
Analysis of Algorithms
Fall 2015
Exam III

Name: _____

Student ID: _____

Email Address: _____

_____ Check if DEN Student

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	12	
Problem 4	15	
Problem 5	15	
Problem 6	12	
Problem 7	11	
Total	100	

Instructions:

1. This is a 2-hr exam. Closed book and notes
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.

1) 20 pts

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

[TRUE/FALSE]

If P = NP, then all NP-Hard problems can be solved in Polynomial time.

[TRUE/FALSE]

Dynamic Programming approach only works when used on problems with non-overlapping sub problems.

[TRUE/FALSE]

In a divide & conquer algorithm, the size of each sub-problem must be at most half the size of the original problem.

[TRUE/FALSE]

In a 0-1 knapsack problem, a solution that uses up all of the capacity of the knapsack will be optimal.

[TRUE/FALSE]

If a problem X can be reduced to a known NP-hard problem, then X must be NP-hard.

[TRUE/FALSE]

If SAT \leq_P A, then A is NP-hard.

[TRUE/FALSE]

The recurrence $T(n) = 2T(n/2) + 3n$, has solution $T(n) = \theta(n \log(n^2))$.

$$a=2, b=2, f(n)=3n = \Theta(n), n^{\log_b^a} = n$$

[TRUE/FALSE]

Consider two positively weighted graphs $G_1 = (V, E, w_1)$ and $G_2 = (V, E, w_2)$ with the same vertices V and edges E such that, for any edge $e \in E$, we have $w_2(e) = (w_1(e))^2$. For any two vertices $u, v \in V$, any shortest path between u and v in G_2 is also a shortest path in G_1 .



[TRUE/FALSE]

If an undirected graph $G=(V,E)$ has a Hamiltonian Cycle, then any DFS tree in G has a depth $|V| - 1$.

[TRUE/FALSE]

Linear programming is at least as hard as the Max Flow problem.

2) 15 pts

A company makes three models of desks, an executive model, an office model and a student model. Building each desk takes time in the cabinet shop, the finishing shop and the crating shop as shown in the table below:

Type of desk	Cabinet shop	Finishing shop	Crating shop	Profit
Executive	2	1	1	150
Office	1	2	1	125
Student	1	1	.5	50
Available hours	16	16	10	

How many of each type should they make to maximize profit? Use linear programming to formulate your solution. Assume that real numbers are acceptable in your solution.

Variable : x_i is the number of type i desks made by the company.

object : $\max (150x_1 + 125x_2 + 50x_3)$

$$\text{s.t.} \quad \begin{cases} 2x_1 + x_2 + x_3 \leq 16 \\ x_1 + 2x_2 + x_3 \leq 16 \\ x_1 + x_2 + 0.5x_3 \leq 10 \\ x_i \geq 0 \end{cases}$$

3) 12 pts

Given a graph $G=(V, E)$ and a positive integer $k < |V|$. The longest-simple-cycle problem is the problem of determining whether a simple cycle (no repeated vertices) of length k exists in a graph. Show that this problem is NP-complete.

Certificate : a list of nodes

Certifier : check # of nodes is k

check no repeat nodes, except first and last

check every two nodes has an edge

check the last nodes is the first nodes

Claim : HC \leq_p LSC

$G(V, E)$ has a HC iff $G(V, E)$ has LSC of size $|V|$

4) 15 pts

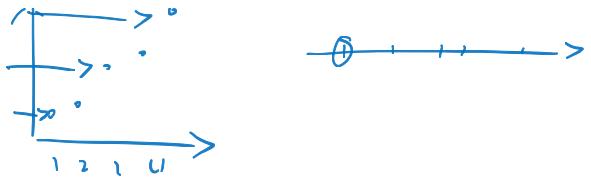
Suppose there are n steps, and one can climb either 1, 2, or 3 steps at a time. Determine how many different ways one can climb the n steps. E.g. if there are 5 steps, these are some possible ways to climb them: (1,1,1,1,1), (1,2,1,1), (3, 2), (2,3), etc. Your algorithm should run in linear time with respect to n . You need to include your complexity analysis.

$$OPT(i) = OPT(i-1) + OPT(i-2) + OPT(i-3)$$

base case : $OPT(0) = 1$
 $OPT(1) = 1$

$$OPT(2) = 2$$

$$OPT(3) = 4$$



5) 15 pts

We'd like to select frequencies for FM radio stations so that no two are too close in frequency (creating interference). Suppose there are n candidate frequencies $\{f_1, \dots, f_n\}$. Our goal is to pick as many frequencies as possible such that no two selected frequencies f_i, f_j have $|f_i - f_j| < e$ (for a given input variable e). Design a greedy algorithm to solve the problem. Prove the optimality of the algorithm and analyze the running time.

Sort frequencies $\uparrow F_1, \dots, F_n$

Check each F_i in this order, pick the first one and delete all the next F_i s that are within e .

Proof: Suppose an opt. sol D

for each frequency $F(s_i) \leq F(D_i)$

for the last frequency we pick $F(s_m) \leq F(D_m)$

Claim: ^{the} opt. sol. can not have D_{m+1} .

if D_{m+1} exist $F(D_{m+1})$ is compatible with $F(D_m)$ and $F(s_m)$

then our sol. must pick D_{m+1}

\therefore our sol. has the same # of frequencies

NPC

$$Q \leq_p S \leq_p R$$

6) 12 pts

Let S be an NP-complete problem, and Q and R be two problems whose classification is unknown (i.e. we don't know whether they are in NP, or NP-hard, etc.). We do know that Q is polynomial time reducible to S and S is polynomial time reducible to R . Mark the following statements True or False **based only on the given information, and explain why.**

(i) Q is NP-complete

$$\text{if } P = NP, \quad Q \in NPC$$

$$\text{if } P \neq NP, \quad Q \in NPC \text{ or } P$$

(ii) Q is NP-hard

$$\text{if } P = NP \quad Q \in NP \text{ hard}$$

$$\text{if } P \neq NP \quad Q \notin NP \text{ hard}$$

(iii) R is NP-complete

$$R \in NPC \text{ or } NP \text{ Hard}$$

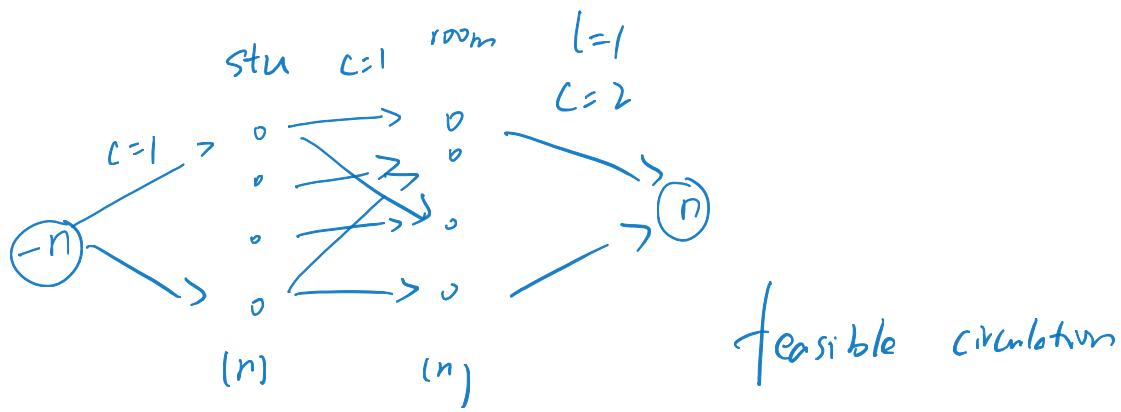
(iv) R is NP-hard

True

7) 11 pts

Consider there are n students and n rooms. A student can only be assigned to one room. Each room has capacity to hold either one or two students. Each student has a subset of rooms as their possible choice. We also need to make sure that there is **at least one student assigned to each room**.

Give a polynomial time algorithm that determines whether a feasible assignment of students to rooms is possible that meets all of the above constraints. If there is a feasible assignment, describe how your solution can identify which student is assigned to which room.



Additional Space

Additional Space