Final Exam Rubric

Problem 1 (15 pts)

Prove or disprove that the following problem is NPC.
Given an undirected graph G = (V, E) with positive edge weights, find a simple cycle (a cycle that has no duplicate nodes) and has a total edge cost of at least D.

Answer: Reduce Hamilton cycle to this problem.
Prove that the problem is in NP:
Certificate: An ordered set of nodes on the cycle with total cycle cost of D (2 pts)
Certifier: Same as those used in TSP. Need to elaborate… (3 pts)

Prove that the problem in NP-hard:
Given an instance G of a Hamilton cycle problem (1 pt), we build G' same as the graph in the Hamilton cycle problem but where all weights are one. We then ask the blackbox to see if there is a simple cycle of cost at least n (number of nodes in G'). (3 pts)

Proof:

A) If we have a Ham Cycle in G, we can find a cycle of cost n in G' using the same set of edges used in the Ham Cycle in G. (3 pts)

B) If we have a cycle of cost n in G' we can find a Ham Cycle in G using the same set of edges used in the cycle. (3 pts)

Rubric:

+4: Showed the whole process of proof, although it is wrong.

1. These 4 scores will only give to students who totally wrong proof the problem is in NP-Hard.
2. The whole process means that you should mention the certificate, certifier, provide the construction process of G' and provide the proof process.

Problem 2 (15 pts)

In a prison, each cell is being protected and monitored by a set of security cameras. A cell might be monitored by more than one security camera and a security camera

can be also monitoring more than one cell. This year they are tight on budget and they want to reduce the electricity usage and maintenance cost by turning off some of the cameras but they also want to make sure that each cell is being monitored by at least one camera.

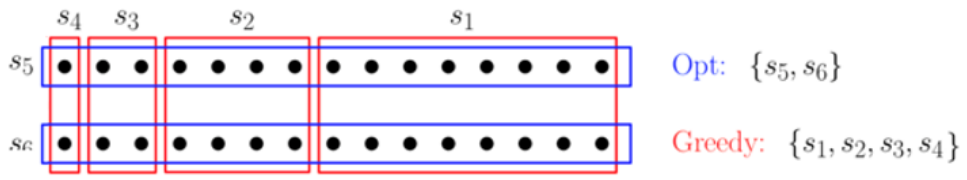
a) (5 pts)

Consider the following algorithm to choose those cameras that need to stay on:

> *At each stage, select the camera that covers the greatest number of uncovered cells.*

Show a counter example to prove that this algorithm does not always give the optimal solution.

Solution:



Rubric:

-5 wrong example

b) (10 pts)

Prove that if we used the above algorithm as an approximation algorithm, this approximation algorithm has no constant approximation ratio > 1.

In other words, if the minimum number of cameras needed is C*, prove that there is no constant ρ (ρ > 1), where we can guarantee our greedy algorithm to achieve an approximate solution with total number of cameras <= ρ C*.

Solution:

You can imagine that in the previous example, after S1, all the remaining sets (except for s5 and s6) are of size 2. Then the value of the optimal solution will be 2 and the value of the approximate solution will be arbitrarily large as you increase the number of cells.

Rubric:
You get full credit for extending the counter example in part (a) of your solution to prove that no such a constant exists.

-5 if the example is correct but not enough explanation is provided.

Problem 3 (10 pts)

Remember Justin and Randall who work at a restaurant near USC? Here is a reminder about how they split their tips:

> *Customers who come to the restaurant usually leave tips (in a form of a currency note) for both of them in a tip jar. Given that there are n notes and every note have a positive value (not necessary the same value for each note) written on it. At the end of their workday, they arrange the notes arbitrarily from left to right on a table (this row of notes is not necessarily sorted). They play the following game to split the tip money: they take turns to play and at each turn, the player chooses either the leftmost note or the rightmost note and takes it. Justin is greedy and always plays using the following strategy: "If the leftmost note has value larger than the rightmost note, then take the leftmost note. Otherwise, take the rightmost note." Construct a dynamic programming algorithm that determines the plays for Randall such that the tip money he gets is maximized. Assume n is even and Randall plays the first turn. We also assume that the values for the notes are given to us—in order—from left to right in an input array v[1..n], i.e. v[1] is the value of the leftmost note and v[n] is the value of the rightmost note.*

The restaurant has now hired a busboy to help set and clear tables and Justin and Randall have decided that they want to change their game the following ways:

- Justin plays the first turn
- The last two notes left on the table will go to the busboy

A) Assuming that is the maximum amount of money that Randall can get from the remaining note sequence , write the recurrence relation for subproblems (5 pts)

Solution. If v(i) > v(j) OPT(i,j) = max{v(i+1)+OPT(i+2,j), v(j)+OPT(i+1,j-1)}.

Otherwise, OPT(i,j) = max{v(i)+OPT(i+1,j-1), v(j-1)+OPT(i,j-2)}.

B) How will you initialize the OPT array? You do not need to write the complete pseudocode. (3 pts)

Base case: OPT(i,i+1) = 0 for i= 1,2,…,n-1

C) [True/False]

To find the value of the optimal solution for this problem a minimum of $\theta(n^2)$ memory is required.

Rubrics:

Full credit for correct answers. 0 for wrong, incomplete or illegible answers.

Problem 4 (15 pts)

Suppose that you are managing a car dealership, and your job is to connect sales agents with the right sellers and buyers. There are k sales agents, *n* sellers each selling a car, and *m* buyers each buying a car. Each buyer has either a low, medium or high credit history. Let us assume that the only factor which matters is the credit history of buyers and the credit history requirement of sellers for a deal to be successful according to the following rules:
(a) High credit buyers can buy from sellers with low, medium, or high credit requirements
(b) Medium credit buyers can only buy from sellers with low and medium credit requirements
(b) Low credit buyers can only buy from sellers with low credit requirements

To make things fair for sales agents, you want to spread buyers with different credit ratings across different sales agents the following way: Each sales agent i can do at most

H(i) deals with high credit buyers
M(i) deals with medium credit buyers
L(i) deals with low credit buyers

Where H(i), M(i), L(i) are positive integers. Give a network flow algorithm to find the assignment of buyers to sales agents such that it results in the maximal number of possible deals.

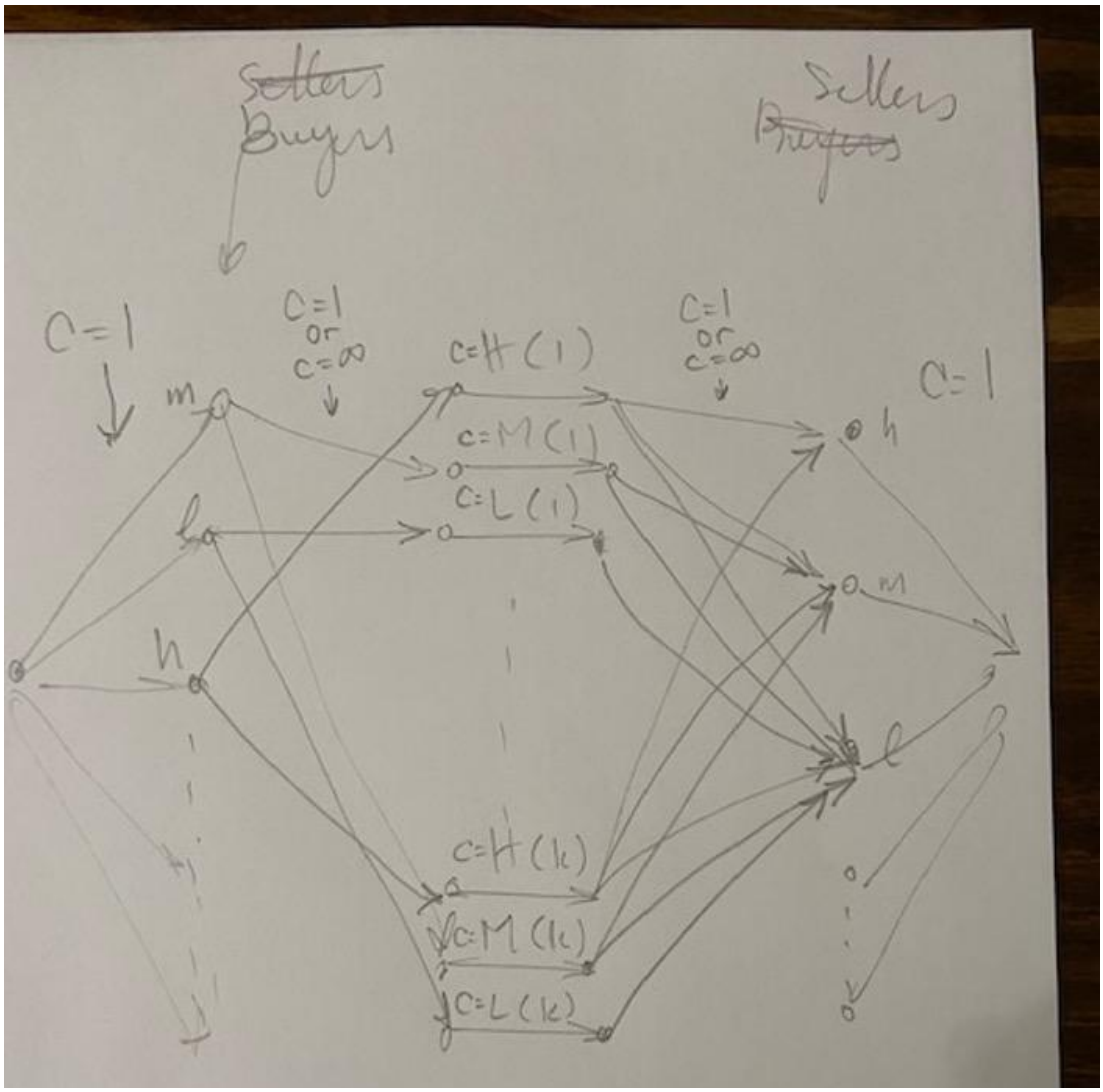
Solution 1:
We define the network as follows:
Nodes:

- A source node s and sink node t.
- A set of n nodes for buyers called X
- A set of m nodes for sellers called Y
- A set of 3k nodes called R where each agent a_i is represented with three nodes a_li, a_mi, a_hi.
- Define R' similar to R.

Edges:

- Connect s to all the nodes in X with capacity 1.
- Connect t to all the nodes in Y with capacity 1.
- Connect buyers according to their credit to the nodes in R. For example if a buyer is high credit, connect it to all a_hi, etc.
- Connect sellers according to the credit rules to the nodes in R'. For example, if a seller is medium credit, connect them to the nodes with medium and low credits.
- Connect each agent node in R to their corresponding node in R': a_li to a'_li, a_mi to a'_mi, a_hi to a'_hi with capacity L(i), M(i) and H(i).

Run max flow.



Solution 2.

We define the network as follows:

Nodes:

- A source node s and sink node t.
- A set of n nodes for buyers called X
- A set of m nodes for sellers called Y
- A set of k nodes for agents called R
- 3 credit nodes as M_b, H_b, and L_b

Edges:

- Connect s to the nodes is set A with capacity inf.
- Connect t to the sellers with capacity 1.
- Connect an agent a_i to M_b, H_b and L_b with capacity M(i), H(i) and L(i).
- Connect buyers according to their credit to the nodes in R. For example if a buyer is high credit, connect it to all a_hi, etc.
- Connect a buyer to a seller if according to the rules, they can trade with each other.

Run MaxFlow Algorithm.

Rubrics:
5 pts: Defining the correct sets of nodes
7 pts: Defining the correct sets of connection and capacities between the nodes you have defined
2 pts: Explaining an algorithm to solve the problem with your network.

Note1. If your solution in any way does not consider 3 (or 3k) nodes defining different credits you were given 5-7 pts.

Note2. If you have considered 3k nodes but the connections are not correct, you were given 8-10 pts.

Note3. If you have considered 3k nodes, the connections between the agents are correct but you have not considered the credit rules between buyers and sellers, 3 points were deducted.