CS570

Analysis of Algorithms
Spring 2011
Exam II

Name: ______________________
Student ID: __________________

DEN Student ____YES ____NO

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

2 hr exam
Close book and notes

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]
If a problem can be solved by dynamic programming, then the problem can be solved in polynomial time

[TRUE/FALSE]
If a problem can be solved by dynamic programming, then it can always be solved by exhaustive search.

[TRUE/FALSE]
There exist some problems that can be solved by dynamic programming, but cannot be solved greedy algorithm.

[TRUE/FALSE]
If f is a maximum s-t flow in G, then for all edges e out of s, we have $f(e) = c_e$.

[TRUE/FALSE]
Given a graph G, if we select an edge e, and increase its capacity by 1, then the maximum flow of the new flow network G' can at most increase by 1.

[TRUE/FALSE] Not Sure
Given any graph G, and a natural number k. Suppose x, y, z are three vertices of G. If there are k edge-disjoint paths from x to y, and k edge-disjoint paths from y to z, then we also have k edge-disjoint paths from x to z.

[TRUE/FALSE]
If you have non integer edge capacities, then you may have an integer max flow.

[TRUE/FALSE]
In the Ford–Fulkerson algorithm, choice of augmenting paths can affect the number of iterations.

[TRUE/FALSE]
Sequence alignment problem between sequence X and sequence Y can be solved using dynamic programming in $O(mn)$ when $|X| = m$ and $|Y| = n$.

[TRUE/FALSE]
The best time complexity to solve the max flow problem can be better than $O(Cm)$ where C is the total capacity of the edges leaving the source and m is the number of edges in the network.

2) 15 pts

Suppose that you are given an n × n checkerboard and a checker. You must move the checker from the bottom edge of the board to the top of the board according to the following rule. At each step you may move the checker to one of three squares:

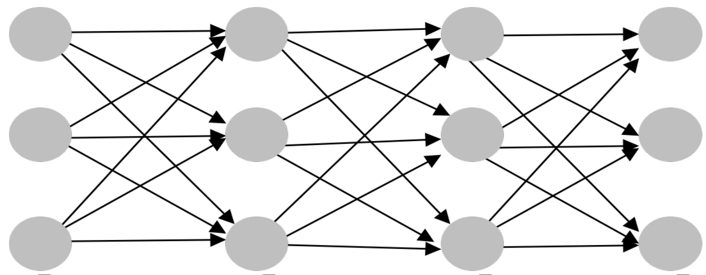
1. the square immediately above,
2. the square that is one up and one to the left (but only if the checker is not already in the leftmost column),
3. the square that is one up and one to the right (but only if the checker is not already in the rightmost column).

Each time you move from square x to square y, you receive p(x, y) dollars. You are given p(x, y) for all pairs (x, y) for which a move from x to y is legal. Do not assume that p(x, y) is positive.

Give an algorithm that figures out the set of moves that will the checker gathering somewhere along bottom edge to somewhere along the top edge while along the bottom edge as a starting point and any square along the top edge as a destination in order to maximize the number of dollars gathered along the way. What is the running time of your algorithm?

3) 20 pts

You are at island I_0 and want to have vacation at island I_n, before arriving at I_t, you need to rent a boat and go to island $I_1, I_2, \ldots I_{n-1}$. At each island $I_0, I_1, I_2, \ldots I_n$, there are m docks where you can land in the island, and you can start from any dock of I_0. Also, you can land on any dock k of island I_{i+1} from any dock j of island I_i with the cost of C_{jk}. Once you land on a dock in an island, you can only start from the SAME dock and continue your trip to the next island. Your task is to figure out the cheapest way to arrive at I_n from I_0 using dynamic programming.



a) Consider the following example n=3, m = 3, and the cost is denoted in the table below.

	k=1	k=2	k=3
j=1	3	5	8
j=2	2	20	6
j=3	10	7	4

Let OPT(i, j) be the cheapest cost from I_0 to dock j of island I_i. Fill out the table below.

	j = 1	**j = 2**	**j = 3**
i= 0			
i= 1			
i= 2			
i= 3			

b) Continuing with the example given in part (1), what is the minimum cost to reach I_3, and what are the corresponding landing docks along the trip? (you need to specify the docks used on I_0, I_1, I_2 and I_3)

c) Continuing from part (1), write down the **recurrence equation** that expresses opt(i, j) in terms of the optimal solution to sub-problems. Make sure to include all boundary conditions.

d) Continuing from part (3), what's the time complexity of the algorithm? You only need to give a formula, no need for justification.

4) 20 pts

N departments are invited to a conference. Department i has d_i members. There are M rooms at the conference, and room j can sit t_j people. To maximize academic interaction, members of the same department are not allowed to sit in the same room. Your task is to formulate this as a basic network flow problem in order to determine if there is a feasible room arrangement for the conference, which means every department member is assigned to a room and no two members of the same department are sitting in the same room; it's okay to have empty rooms. Answer the following questions:

a- How many nodes do you need and what does each type of node correspond to

b- How many edges do you need and what does each type of edge correspond to

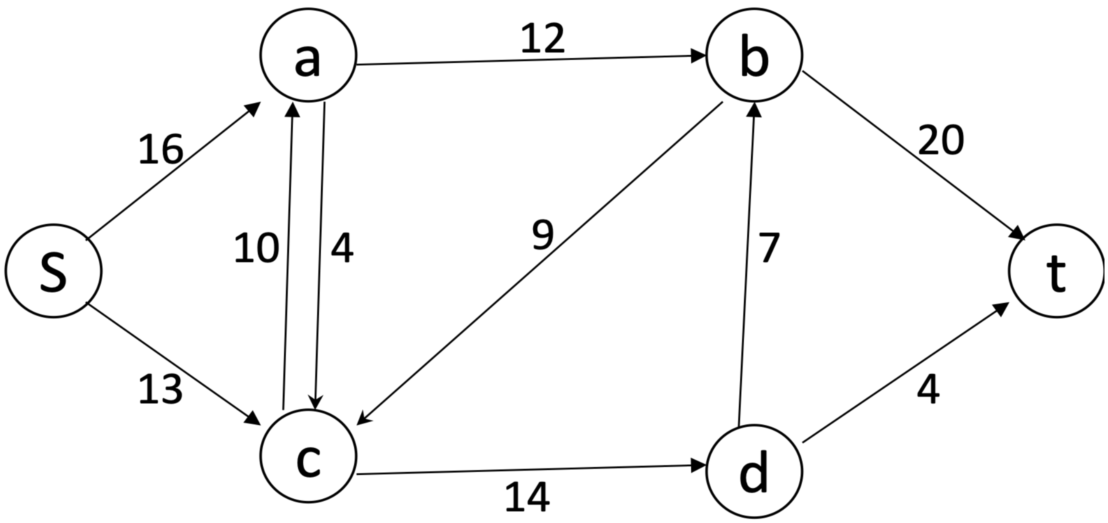
c- What is the capacity of each edge.

Cont'd on next page…

d- Draw the flow network based on part (1)-(3)

e- How do you determine if there is a feasible room arrangement to the problem?

5) 20 pts
Given the flow network below



a) Find a maximum flow from the source node S to the sink node t using the Ford Fulkerson algorithm. You need to show all your iterations.