CS570

Analysis of Algorithms

Spring 2005

Final Exam

Name: ________________________

Student ID: ___________________

1- 10 pts

a) The NP-complete problem X has a polynomial time 2-approximation algorithm A. X can be reduced to problem Y. Can A be used to find a 2-approximation to Y? Answer Yes, No, or Sometimes. Then explain your answer.

b) The NP-complete problem X has a polynomial time 2-approximation algorithm A. Problem Y can be reduced to X. Can A be used to find a 2-approximation to Y? Answer Yes, No, or Sometimes. Then explain your answer.

2- 20 pts

You are given an m by n matrix of real numbers such that the sum of the values in each row and each column are exact integer numbers. Design an algorithm that rounds all the entries in the matrix up or down to the closest integer value without affecting the row or column sums. An example of such a transformation is shown below. Analyze the complexity of your solution.

Example: Input is the 2x3 matrix shown in the shaded area.

	CS570	CS571	CS585	Total
Men	**16.6**	**18.1**	**21.3**	56
Women	**10.4**	**17.9**	**15.7**	44
	27	36	37	100

Output is the 2x3 matrix shown in the shaded area.

	CS570	CS571	CS585	Total
Men	**16**	**19**	**21**	56
Women	**11**	**17**	**16**	44
	27	36	37	100

Hint: Show how this problem can be reduced to one of the variants of the network flow problem we have studied.

3- 20 pts

Assume that you have an n by n checkerboard. You must move a checker from the bottom left corner square of the board to the top right corner square. In each step you may either 1) move the checker up one square, or 2) move the checker diagonally one square up and to the right, or 3) move the checker right one square. If you move a checker from square x to square y you get $p(x, y)$ dollars. You are told all of the $p(x, y)$ a priori. The $p(x, y)$ may be negative, zero or positive. You want to get as much money as possible. Give an efficient algorithm, for this problem. Analyze the complexity of your algorithm.

4- 10 pts

Prove or disprove the following: Given a graph G=(V,E) with edge costs C_e, we can always swap any 2 edge costs without affecting the cost of the Min Spanning Tree of G?

5- 10 pts

a) We have seen a polynomial time solution to the “integer” network flow problem using the Ford-Fulkerson algorithm. Can we also apply linear programming to solve this problem in polynomial time? Explain your answer.

b) Can you apply linear programming to solve the network flow problem when capacities are allowed to take on real numbers? If no, explain why not. If yes, briefly explain how the linear programming problem can be formulated.

6- 20 pts

The input to the Fixed Hamiltonian path problem is an undirected graph *G* and two vertices *x* and *y* in *G*. The problem is to determine if there is a **simple** path between *x* and *y* in *G* that spans **all** the vertices in *G*. A path is simple if it doesn't include any vertex more than once. Show that the Fixed Hamiltonian path problem is NP-complete