CS570

Analysis of Algorithms

Fall 2010

Final Exam

Name: ______________________

Student ID: __________________

____Monday Section ____Friday Section ____DEN Section

	Maximum	Received
Problem 1	20	
Problem 2	20	
Problem 3	20	
Problem 4	10	
Problem 5	20	
Problem 6	10	
Total	100	

2 hr exam
Close book and notes

If a description to an algorithm is required please limit your description to within 200 words, anything beyond 200 words will not be considered.

1) 20 pts

Mark the following statements as **TRUE** or **FALSE** (or **UNKNOWN** if given as an option). No need to provide any justification.

[TRUE/FALSE]
In a flow network, there always exists a maximum flow without cycles carrying positive flow.

[TRUE/FALSE]
In a directed graph with at most one edge between each pair of vertices, if we replace each directed edge by an undirected edge, the maximum flow value remains unchanged.

[TRUE/FALSE]
In the worst case, merge sort runs in $O(n^2)$ time.

[TRUE/FALSE]
Fibonacci heaps can be used to make Dijkstra's algorithm run in O(|E| + |V| log|V|) time on a graph G=(V,E).

[TRUE/FALSE]
If some of the edge weights in a graph are negative, the shortest path from s to t can be obtained using Dijkstra's algorithm by first adding a large constant C to each edge weight, where C is chosen large enough that every resulting edge weight will be nonnegative.

[TRUE/FALSE/UNKNOWN]
A problem can be both NP-complete and undecidable.

[TRUE/FALSE]
If a problem A is in P and $A \leq_P B$ for some other problem B, then B is in P as well.

[TRUE/FALSE/UNKNOWN]
If there is a polynomial time algorithm for some problem in NP, then all problems in NP can be solved in polynomial time.

[TRUE/FALSE]
If a problem A is NP-hard and $A \leq_P B$ for some other problem B, then B is NP-hard as well.

[TRUE/FALSE/UNKNOWN]
It is proven that there is a problem that belongs in P but it is not NP-complete.

2) 20 pts

The shortest common supersequence:

The input comprises two strings of characters, S [1 : n] and T [1 : n] stored in arrays. The problem is to determine the length r of a shortest string U [1 : r] such that S and T are both subsequences of U . e.g. Suppose S = ABAC and T = ACDC . Then U = ABCDAC is a shortest common superstring. Give an efficient algorithm to find the length of U.

3) 20 pts

Prove that the following problem--Set Packing--is NP-complete. Given m sets S_1 , S_2 , . . . , S_m, find as many of these sets as possible such that no selected pair of sets intersects. That is, find the largest set $C \subseteq \{1, . . . , m\}$ such that for all $i, j \in C$, we have $S_i \cap S_j = \emptyset$. First phrase the problem as an equivalent decision problem, and then prove the decision problem is NP-complete.

4) 10 pts
   If there is a polynomial-time 1/2-approximation algorithm for the weighted vertex cover problem then there is a polynomial-time 1/2-approximation algorithm for the Independent Set problem. Prove or disprove the statement.

5) 20 pts

We define a *most vital arc* of a network as an arc whose deletion causes the largest decrease in the maximum *s*-*t*-flow value. Let f be an arbitrary maximum *s*-*t*-flow. Either prove the following claims or show through counterexamples that they are false:

(a) A most vital arc is an arc *e* with the maximum value of *c(e)*.

(b) A most vital arc is an arc *e* with the maximum value of *f(e)*.

(c) A most vital arc is an arc *e* with the maximum value of *f(e)* among arcs belonging to some minimum cut.

(d) An arc that does not belong to some minimum cut cannot be a most vital arc.

(e) A network might contain several most vital arcs.

.

6) 10 pts

Suppose you have an undirected graph with weighted edges, and perform a depth first search, such that the edges going out of each vertex are always explored in order by weight, smallest first. Is the depth first search tree resulting from this process guaranteed to be a minimum spanning tree? Explain why, if it is, or, if it isn't, provide a counterexample.

Additional Space

Additional Space