CS570
Analysis of Algorithms
Spring 2012
Exam I

Name: ______________________

Student ID: __________________

____12:30 – 1:50 session   ____ 2:00 – 3:20 session

	Maximum	Received
Problem 1	20	
Problem 2	20	
Problem 3	20	
Problem 4	10	
Problem 5	10	
Problem 6	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]
Dijkstra's algorithm solves the Minimum Spanning Tree problem.

[TRUE/FALSE]
Let e be a maximum-weight edge on some cycle of $G = (V, E)$. Then there is a minimum spanning tree of $G' = (V, E - \{e\})$ that is also a minimum spanning tree of G.

[TRUE/FALSE]
Let e be a minimum-weight edge in a graph G. Then e must belong to some minimum spanning tree of G.

[TRUE/FALSE]
Given a directed graph G with n vertices, any BFS tree corresponding to graph G will have n-1 edges.

[TRUE/FALSE]
Given an undirected graph G with n vertices, any BFS tree corresponding to graph G will have n-1 edges.

[TRUE/FALSE]
A constant $n_0 \geq 1$ exists such that for any $n \geq n_0$, there is an array of n elements such that insertion sort runs faster than merge sort on that input.

[TRUE/FALSE]
A constant $n_0 \geq 1$ exists such that for any $n \geq n_0$, there is an array of n elements such that insertion sort runs slower than merge sort on that input.

[TRUE/FALSE]
The array [20, 15, 18, 7, 9, 5, 12, 3, 6, 2] forms a MaxHeap.

[TRUE/FALSE]
Having done two BFS and DFS searches starting from the same point s in graph G, the DFS tree is never as the same as the BFS tree.

[TRUE/FALSE]
A greedy algorithm always makes the choice that looks best at the moment.

2) 20 pts

a- Show that a graph has a unique minimum spanning tree if, for every cut of the graph, there is a unique light edge (i.e., a unique edge of smallest cost) crossing the cut.
Definition: a cut in a graph divides the nodes of a graph into two sets. For example, in class we talked about the node sets S and V-S when discussing some MST algorithms. The boundary between S and V-S is an example of a cut.

b- Show that the converse is not true, i.e. it is possible for a graph to have non-unique minimum cost edges crossing a cut and still only have one unique MST.

3) 20 pts

You are given n events where each takes one unit of time. Event i will provide a profit of g_i dollars ($g_i > 0$) if started at or before time t_i where t_i is an arbitrary real number. All events can start as early as time 0, but events cannot be scheduled at the same time. Give the most efficient algorithm you can to find a schedule that maximizes the profit.

Note: If an event is not started by t_i then there is no benefit in scheduling it at all.

4) 10 pts

Consider two positively weighted graphs $G = (V, E, w)$ and $G' = (V, E, w')$ with the same vertices V and edges E such that, for any edge e in E, we have $w'(e) = w(e)^2$.

Prove or disprove: For any two vertices u, v in V, any shortest path between u and v in G' is also a shortest path in G.

5) 10 pts

Assume that A is a very large unsorted array of integers. Describe an algorithm that picks the largest *m* integers in A, where *m* is small relative to the size of the array (*n*). The time complexity should be less than $O(nlgn)$, i.e. sorting the array is not going to be fast enough. Also, the amount of additional memory that you are given is O(1).

6) 20 pts

a- Describe how Strassen's algorithm is capable of reducing the complexity of dense matrix multiplication. You do not have to provide exact formulas. You only need to describe the approach and to do complexity analysis on the resulting algorithm.

b- Solve the following recurrences by giving tight Θ-notation bounds. You do not need to justify your answers, but any justification that you provide will help when assigning partial credit.

i. $T(n)=4T(n/2)+n^2 logn$

ii. $T(n)=8T(n/2)+nlogn$

iii. $T(n)=(6006)^{1/2}*T(n/2)+n^{\sqrt{6006}}$

Additional Space

Additional Space