CS570
Analysis of Algorithms
Spring 2014
Exam I

Name: ______________________
Student ID: __________________

____On Campus ____DEN

	Maximum	Received
Problem 1	20	
Problem 2	16	
Problem 3	16	
Problem 4	16	
Problem 5	16	
Problem 6	16	
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]
The number of cycles in a bipartite graph is never odd.

[TRUE/FALSE]
Every tree is a bipartite graph.

[TRUE/FALSE]
If a weighted undirected graph has two MSTs, then it has a cycle C such that the maximum weight edge in C is not unique.

[TRUE/FALSE]
If T is an MST for a weighted undirected graph, then T remains being an MST even if the edge weights are doubled.

[TRUE/FALSE]
Given a binary max-heap with n elements**,** the time complexity of extracting the smallest element from the heap is O(log n)

[TRUE/FALSE]
If f, g, and h are positive increasing functions with f in $O(h)$ and g in $\Omega(h)$, then the function $f+g$ must be in $\Theta(h)$.

[TRUE/FALSE]
The array [20 15 18 7 9 5 12 3 6 2] forms a binary max-heap.

[TRUE/FALSE]
The number of spanning trees in a fully connected graph with n vertices goes up exponentially with respect to n.

[TRUE/FALSE]
If the edges in a connected graph all have unit costs, then the shortest path between two nodes in found faster using BFS than it is using Dijkstra's algorithm.

[TRUE/FALSE]
Given a problem with input of size n, a solution with O(n) time complexity always costs less in computing time than a solution with O(n^2) time complexity.

2) 16 pts

You are given a weighted directed graph $G = (V,E)$ and the shortest path **distances** $\delta(s, u)$ from a source vertex s to every other vertex in G. However, you are not given $\pi(u)$ (the predecessor pointers). With this information, give an algorithm to find a shortest path from s to a given vertex t in $O(|V| + |E|)$ time.

3) 16 pts

   You are given n jobs each with a known start and end time. There are n identical processors. Two jobs with overlapping running times cannot be assigned to the same processor. Describe an algorithm to assign jobs to processors such that the number of processors utilized is minimized.

4) 16 pts

The police department in a city has made all streets one-way. The mayor contends that there is still a way to drive legally from any intersection in the city to any other intersection, but the opposition is not convinced. A computer program is needed to determine whether the mayor is right. However, the city elections are coming up soon, and there is just enough time to run a *linear-time* algorithm.

i) Formulate this as a graph problem and design a linear-time algorithm. Explain why it can be solved in linear time.

ii) Suppose it now turns out that the mayor's original claim is false. She next makes the following claim to supporters gathered in the Town Hall: “If you start driving from the Town Hall (located at an intersection), navigating one-way streets, then no matter where you reach, there is always a way to drive legally back to the Town Hall.” Formulate this claim as a graph problem, and show how it can also be verified in linear time.

5) 16 pts

Consider the following modification to Dijkstra's algorithm for single source shortest paths to make it applicable to directed graphs with negative edge lengths.

If the minimum edge length in the graph is $-w < 0$, then add $w+1$ to each edge length thereby making all the edge lengths positive. Now apply Dijkstra's algorithm starting from the source s and output the shortest paths to every other vertex.

Does this modification work ? Either prove that it correctly finds the shortest path starting from s to every vertex or give a counter example where it fails.

6) 16 pts

Design a data structure that has the following properties:

- Find median takes $O(1)$.
- Extract-Median takes $O(\log n)$.
- Insert takes $O(\log n)$.
- Delete takes $O(\log n)$.

(**Hint:** Your Data Structure should use a min-heap and a max-heap simultaneously where half of the elements are in the max-heap and the other half are in the min-heap).

a) Describe how your data structure will work.

b) Give an $O(\log n)$ algorithm for inserting a new element in the data structure.

c) Give an $O(\log n)$ algorithm for Extract-Median a new element in the data structure.

.

Additional Space