CS570
Analysis of Algorithms
Spring 2011
Exam I

Name: ________________________

Student ID: ____________________

DEN Student ____YES ____NO

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	10	
Problem 4	10	
Problem 5	15	
Problem 6	15	
Problem 7	15	
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]
DFS tree is never the same as a BFS tree for the same graph and starting point.

[TRUE/FALSE]
If e is a minimum-weight edge in a graph G, it belongs to some minimum spanning tree of G.

[TRUE/FALSE]
The complexity of a divide and conquer algorithm with recurrence equation $T(n) = 3T\left(\frac{n}{4}\right) + \Theta(n^2)$ *is* $\Theta(n^2 \log(n))$.

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

[TRUE/FALSE]
Fibonacci heap can be a binary heap.

[TRUE/FALSE]
Consider an instance of the Stable Matching Problem in which there exists a man *m* and a woman *w* such that *m* is ranked last on the preference list of *w* and *w* is ranked last on the preference list of *m*, then in every stable matching *S* for this instance, the pair (*w*, *m*) belongs to *S*.

[TRUE/FALSE]
Suppose that in an instance of the original Stable Marriage problem with n couples, there is a man M who is first on every woman's list and a woman W who is first on every man's list. If the Gale-Shapley algorithm is run on this instance, then M and W will be paired with each other.

[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.

[TRUE/FALSE]
The worst-case time complexity of merge sort is O(nlgn)

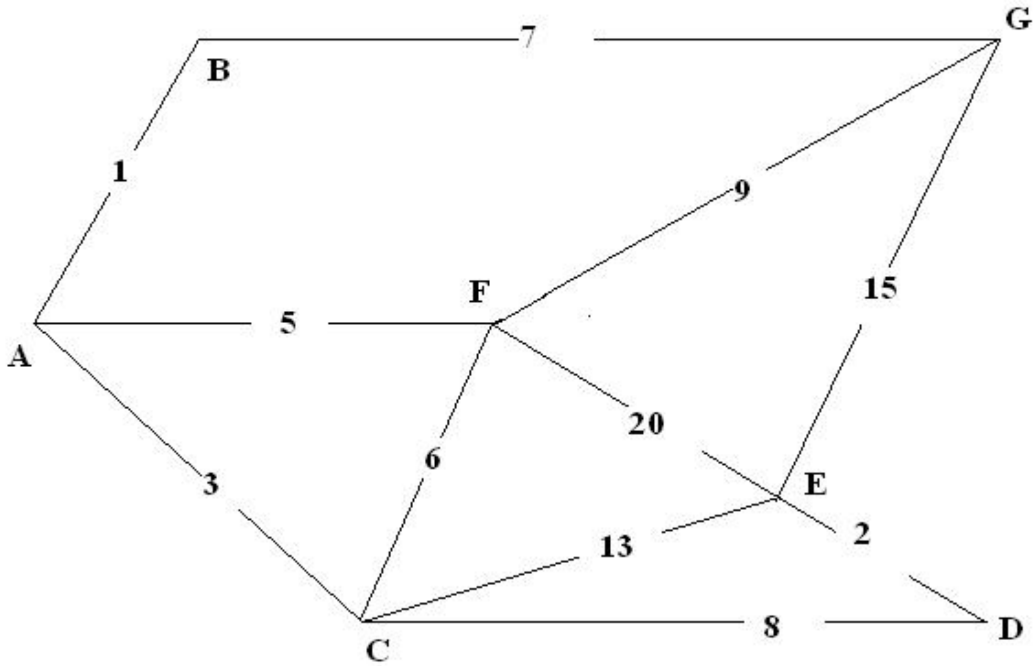
[TRUE/FALSE]
Both BFS and DFS can be used to find all connected components of a graph.

2) 15 pts

Given an array with n numbers, design an algorithm that finds the maximum and minimum among the numbers. Assume $n=2^i$, where i is a natural number. You algorithm must do exactly $3/2n$ - c comparisons between numbers, where c is a constant.

3) 10 pts

Given the graph below, you are required to run Kruskal's algorithm to find a MST in the graph.



(1) List the edges selected by the algorithm. The order of the edges selected should strictly follow the algorithm.

(2) What are (is) the edge(s) that you inspected during the execution of the algorithm but rejected as an edge for the MST?

4) 10 pts

The recurrence $T(n) = 7T\left(\frac{n}{2}\right) + n^2$ describes the running time of an algorithm A. A competing algorithm A' has a running time of $T'(n) = aT'\left(\frac{n}{4}\right) + n^2$. What is the largest integer value for a such that A' is asymptotically faster than A?

.

5) 15 pts

Suppose you are given a set $S = \{a_1, a_2, \dots, a_n\}$ of tasks. Where task a_i requires p_i units of processing time to complete, once it has started. You have one computer on which to run these tasks, and the computer can run only one task at a time. Let c_i be the ***completion time*** of task a_i, that is, the time at which the task a_i completes processing. Your goal is to minimize the average completion time, that is, to minimize $(\frac{1}{n})\sum_{i=1}^{n} c_i$. For example, suppose there are two tasks, a_1 and a_2, with $p_1 = 3$ and $p_2 = 5$, and consider the schedule in which a_2 runs first, followed by a_1. Then $c_2 = 5$, $c_1 = 8$, and the average completion time is $\frac{5+8}{2} = 6.5$.

Give an algorithm that schedules the tasks so as to minimize the average completion time. Each task must run non-preemptively, that is, once task a_i is started, it must run continuously for p_i units of time. Prove that your algorithm minimizes the average completion time, and state the running time of your algorithm.

6) 15 pts

Given a sorted array of n integers that has been rotated an unknown number of times, give an O(log n) algorithm that finds an element in the array.

EXAMPLE of array rotation:

Original sorted array a = [1, 3, 5, 7, 11]

After first rotation a’ = [3,5,7,11,1]

After second rotation a’’ = [5, 7, 11, 1, 3]

7) 15 pts

We are given a directed graph $G = (V, E)$ on which each edge $(u, v) \in E$ has an associated value $r(u, v)$, which is a real number in the range $0 \leq r(u, v) \leq 1$ that represents the reliability of a communication channel from vertex u to vertex v. We interpret $r(u, v)$ as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between two given vertices.

Additional Space

Additional Space