CS570

Analysis of Algorithms

Spring 2016

Exam I

Name: ______________________

Student ID: __________________

Email Address:________________

_____Check if DEN Student

	Maximum	Received
Problem 1	20	
Problem 2	20	
Problem 3	15	
Problem 4	15	
Problem 5	15	
Problem 6	15	
Total	100	

Instructions:

1. This is a 2-hr exam. Closed book and notes
2. If a description to an algorithm or a proof is required please limit your description or proof to within 150 words, preferably not exceeding the space allotted for that question.
3. No space other than the pages in the exam booklet will be scanned for grading.
4. If you require an additional page for a question, you can use the extra page provided within this booklet. However please indicate clearly that you are continuing the solution on the additional page.

1) 20 pts

Mark the following statements as **TRUE** or **FALSE**. No need to provide any justification.

[TRUE/FALSE]
All stable matchings are perfect matchings.

[TRUE/FALSE]
In class, we showed that choosing requests with earliest finish time will lead to an optimal solution to the interval scheduling problem. An optimal solution can also be achieved by choosing requests with latest start time.

[TRUE/FALSE]
The relaxation step in Dijkstra's shortest path algorithm will have a lower worst case time complexity if we used a Fibonacci heap as opposed to a binary heap.

[TRUE/FALSE]
BFS can be used to find the shortest path between two nodes in any graph as long as the edge costs are all positive.

[TRUE/FALSE]
If function $f=O(n^3)$ and $g=O(n^2)$, then $f/g=O(n)$.

[TRUE/FALSE]
A graph has a unique MST if and only if all its edges have different weights.

[TRUE/FALSE]
A binary heap A has each key randomly increased or decreased by 1. The random choices are independent. We can restore the heap property on A in linear time.

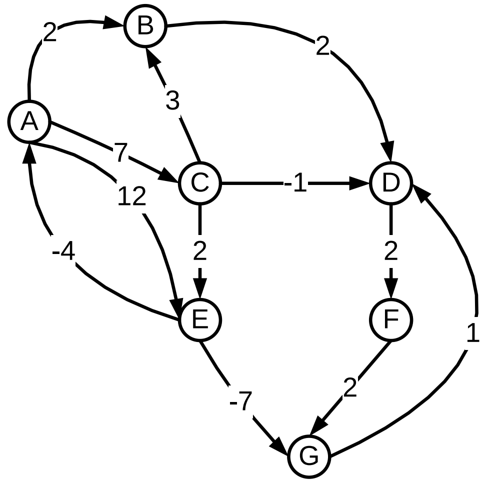
[TRUE/FALSE]
If item A is an ancestor of item B in a heap (used as a Priority Queue) then it must be the case that the Insert operation for item A occurred before the Insert operation for item B.

[TRUE/FALSE]
Prim's algorithm works correctly even when there are negative cost edges in the graph.

[TRUE/FALSE]
Every directed acyclic graph has exactly one topological ordering.

2) 20 pts

Consider the following directed, weighted graph:



a. Even though the graph has negative weight edges, step through Dijkstra's algorithm to calculate *supposedly* shortest paths from *A* to every other vertex. Show your steps in the table below. Cross out old values and write in new ones, from left to right within each cell, as the algorithm proceeds. Also, list the vertices in the order that Dijkstra's finds their shortest path.

Vertex	Found at Step	Distance	Path
A			
B			
C			
D			
E			
F			
G			

b. Dijkstra's algorithm found the wrong path to some of the vertices. For **just the vertices where the wrong path was computed**, indicate *both* the path that was computed and the correct path.

c. List the minimum number of edges that could be removed from the graph so that Dijkstra's algorithm would happen to compute correct answers for all vertices in the remaining graph.

3) 15 pts

You are given a set of n intervals on the x-axis (a line): $[s_1, f_1]$; $[s_2, f_2]$; ……; $[s_n, f_n]$, where the i^{th} interval has left endpoint at $x = s_i$ and its right endpoint at $x = f_i$. Your goal is to select the minimum number of intervals whose union is the same as the union of all intervals. Give an efficient algorithm for this problem, analyze its running time, and prove that it is correct.

4) 15 pts

We have two routines for graph traversal - DFS(G, s) and BFS(G, s) - where G is a graph and s is the starting node in G. These two procedures will create a DFS tree and a BFS tree rooted at s respectively.

CLAIM: If G = (V, E) is a connected, undirected graph then the height of DFS(G, t) tree (rooted at an arbitrary node t) is always larger than or equal to the height of any of the BFS trees created by BFS(G, x).

Either prove this claim is true or provide a counterexample.

Note: we are comparing the height of one DFS tree rooted at t to the heights of |V| BFS trees each rooted at a different node in G.

5) 15 pts

Consider an array *A* containing *n* **distinct** integers. We define a local minimum of *A* to be an *x* such that *x=A[i]*, for some $0 \le i < n$, with $A[i-1] > A[i]$ and $A[i] < A[i+1]$. In other words, a local minimum *x* is less than its neighbors in *A* (for boundary elements, there is only one neighbor to consider). As an example, suppose A = [10, 6, 4, 3, 12, 19, 18]. Then A has two local minima: 3 and 18.

a. Prove that **any** array *A* will have **at least** one local minimum

b. Describe an algorithm using the divide and conquer technique to find **a** local minimum. Note that *A* might have multiple local minima, but you only have to locate and return one.

c. Express a recurrence equation for the running time of your algorithm and solve the recurrence.

6) 15 pts

You are given a graph $G = (V, E)$ with positive edge weights, and a minimum spanning tree $T = (V, F)$ with respect to these weights. Now suppose the weight of a particular edge $e \in E$ is modified from $w(e)$ to a new value $\hat{w}(e)$. You wish to quickly update the minimum spanning tree T to reflect this change, without re-computing the entire tree from scratch. There are four cases. In each case give a linear-time algorithm for updating the tree (if the tree needs to be updated).

(a) $e \in E - F$ and $\hat{w}(e) > w(e)$.
(b) $e \in E - F$ and $\hat{w}(e) < w(e)$.
(c) $e \in F$ and $\hat{w}(e) < w(e)$.
(d) $e \in F$ and $\hat{w}(e) > w(e)$.

Additional Space

Additional Space