CMPSCI 311: Introduction to Algorithms
First Midterm Exam

March 2, 2017.

Name: ______________________________ ID: ____________________

Instructions:

- Answer the questions directly on the exam pages.
- Show all your work for each question. Providing more detail including comments and explanations can help with assignment of partial credit.
- If you need extra space, use the blank pages at the end of the exam.
- No books, notes, calculators or other electronic devices are allowed. Any cheating will result in a grade of 0.
- If you have questions during the exam, raise your hand.

Question	Value	Points Earned
1	10	
2	10	
3	10	
4	10	
5	10	
Total	50	

Question 1. (*10 points*) For each pair of functions f and g, circle the statements that are true. No justification is needed.

1.1 (*2 points*): $f(n) = 10n,\ g(n) = n + 1000$. *Circle all that apply:*

(a) $f(n)$ is $O(g(n))$ (b) $f(n)$ is $\Omega(g(n))$

1.2 (*2 points*): $f(n) = 5 \log n,\ g(n) = (\log n)^2$. *Circle all that apply:*

(a) $f(n)$ is $O(g(n))$ (b) $f(n)$ is $\Omega(g(n))$

1.3 (*2 points*): $f(n) = 5 \log n,\ g(n) = \log(n^2)$. *Circle all that apply:*

(a) $f(n)$ is $O(g(n))$ (b) $f(n)$ is $\Omega(g(n))$

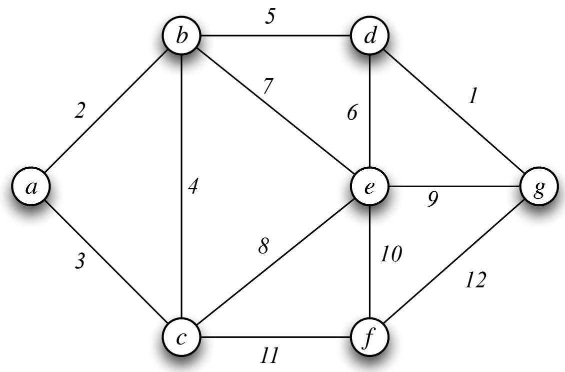
1.4 (*2 points*): $f(n) = \sum_{i=1}^{n} i^3,\ g(n) = n^3$. *Circle all that apply:*

(a) $f(n)$ is $O(g(n))$ (b) $f(n)$ is $\Omega(g(n))$

1.5 (*2 points*): $f(n) = n2^n, g(n) = 3^n/n^3$. *Circle all that apply:*

(a) $f(n)$ is $O(g(n))$ (b) $f(n)$ is $\Omega(g(n))$

Question 2. (*10 points*) Consider the following graph with distinct edge costs.



2.1 (*2 points*): *List the costs of the edges of the minimum-spanning tree in the order they are added by Kruskal's algorithm. Recall Kruskal's algorithm considers edges in a fixed order.*

2.2 (*2 points*): *List the costs of the edges of the minimum-spanning tree in the order they are added by Prim's algorithm starting from node a. Recall that Prim's algorithm maintains a single connected component containing the starting node.*

2.3 (*2 points*): *Recall that Dijkstra's shortest-path algorithm starts with a set* $S = \{s\}$ *of* explored nodes *and defines* $d[s] = 0$*. In each iteration, the algorithm adds some node* v *to* S*. Suppose Dijkstra's algorithm is run with* $s = g$*. List the next two nodes that are added to* S *by the algorithm.*

2.4 (*1 points*): *Suppose a BFS is performed starting at node e. How many layers will the BFS tree have?*

2.5 (*1 points*): *What is the length of the shortest path from a to g?*

2.6 (*1 points*): *Suppose a DFS is performed starting at node a. How deep will the DFS tree be? (Recall that the depth is the maximum number of edges in the path from the root to to any leaf.)*

2.7 (*1 points*): *Is this graph bipartite?*

Question 3. (*10 points*) Indicate whether each of the following statements is TRUE or FALSE. No justification required.

3.1 (*2 points*): *Consider a stable matching instance where every student has a different top-ranked college. There is a unique stable matching in this instance.*

3.2 (*2 points*): *Suppose G is an undirected graph where every node has degree exactly two, and n is odd. Then G is not bipartite.*

3.3 (*2 points*): *Suppose G is an undirected graph with edge costs c_e. Let $(S, V-S)$ be any cut. If e is not the lowest-cost edge across cut $(S, V-S)$, then e does not belong to any minimum spanning tree.*

3.4 (*2 points*): *Suppose G is an undirected graph with edge costs c_e. Let $(S, V-S)$ be any cut. If e is lowest-cost edge across cut $(S, V-S)$, then e belongs to every minimum spanning tree.*

3.5 (*2 points*): *Let T be a minimum spanning tree of an undirected graph G with edge costs c_e. Suppose the edge costs are all increased by one, i.e., $c'_e = c_e + 1$. Then T is still a minimum spanning tree with the new edge costs c'_e.*

Question 4. (*10 points*) Let $G = (V, E)$ be a directed graph where every node has exactly one incoming edge, and there is a node s that has a directed path to every other node.

4.1 (*2 points*): *Draw an example graph with at least 6 nodes that satisfies the conditions stated about G. Identify s in your example.*

4.2 (*2 points*): *Prove that G is not a DAG.*

4.3 (*3 points*): *Suppose you want to carefully delete edges from G to turn it into a DAG. What is the smallest number of edges you need to delete to guarantee that the resulting graph G' is a DAG? Prove that your answer is correct.*

4.4 (*3 points*): *Now give an $O(m+n)$ time algorithm to identify the specific set of edges to delete. This set should be as small as possible and result in a graph G' that is a DAG. Prove that your algorithm is correct.*

Question 5. (*10 points*) You play on a local game show where you will complete n different challenges and try to make as much money as possible. The ith challenge takes t_i minutes to complete, and starts with a value of v_i dollars, but its value decreases by one dollar per minute. In other words, if you complete challenge i at time $f(i)$, you earn $v_i - f(i)$ dollars. Every challenge has starting value $v_i \geq \sum_{i=1}^{n} t_i$, so you will make *some* money from each challenge. You goal is to order the challenges to earn as much as possible.

5.1 (*2 points*): *Suppose* $v_1 = 10, v_2 = 15, v_3 = 20$ *and* $t_1 = 2, t_2 = 5, t_3 = 1$.

What is the optimal ordering?

How much do you earn?

5.2 (*3 points*): *Your friend claims that if all challenges have their value changed to the same number* V*, this will not change the optimal ordering. Is she right? Either prove she is right or provide a counterexample to show that she is wrong.*

5.3 (*2 points*): *One of these orderings is guaranteed to be optimal. Circle the correct one.*

Sort challenges by:

increasing v_i *decreasing* v_i *increasing* t_i *decreasing* t_i

5.4 (*3 points*): *Prove that the ordering you circled always finds an optimal solution.*