CMPSCI 311: Introduction to Algorithms
First Midterm Exam

October 13, 2016.

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 back of a page.
- 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*) Indicate whether each of the following statements is TRUE or FALSE. No justification required.

1.1 (*2 points*): $\sum_{i=1}^{n} i = \Theta(n^2)$.

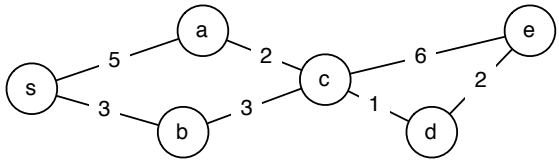
1.2 (*2 points*): *If T is a BFS tree for a graph G and (u, v) is an edge in G that is not in T, then u and v are the same distance from the root of T.*

1.3 (*2 points*): *Given n colleges and n applicants where each college has a rank-ordering of all the applicants (no ties) and every applicant has a rank-ordering of all the colleges (no ties), there can be at most one stable matching.*

1.4 (*2 points*): *Every directed acyclic graph has a node with no incoming edges.*

1.5 (*2 points*): *The recurrence $T(n) = 3T(n/2) + n$ solves to $T(n) = \Theta(n^{\log_2 3})$.*

Question 2. (*10 points*) In the first two parts of this question we consider the following weighted graph G:



2.1 (*4 points*): *For the above graph:*

What is the length of the shortest path from s to e?
How many different spanning trees are there?
How many different minimum spanning trees are there?
What's the weight of the minimum spanning tree?

2.2 (*2 points*): *Recall that Dijkstra's 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 and defines $d[v]$.*

What node is added in the first iteration?
What node is added in the second iteration?

2.3 (*2 points*): *Your friend claims that the order in which the nodes of an arbitrary graph are added to the set of explored nodes when running Dijkstra's algorithm is unchanged if we increment every edge weight by one. Is this always true? Justify your answer.*

2.4 (*2 points*): *Assume edge weights are integers. Another friend claims she can find the shortest path between s and every node by building a BFS tree from s if we first replace every edge with an unweighted path of length equal to the weight of the edge. Is this always true? Justify your answer.*

Question 3. (*10 points*) In this question, we consider analyzing the performance of a specific company in the stock market. Suppose we tracked the stock over a sequence of n days and let $A[t]$ be the value of the stock at the end of the tth day. For each of the following problems, give a brief description of a *simple* algorithm and state the running time of your algorithm. No proof required.

3.1 (*2 points*): *Compute the maximum value of the stock, i.e.,* $\max_{1 \le t \le n} A[t]$.

Running Time:

Brief Description:

3.2 (*2 points*): *Compute the average change from the previous day, i.e.,* $\frac{1}{n} \sum_{t=1}^{n-1} (A[t+1] - A[t])$. **Hint:** *Can you simplify this expression?*

Running Time:

Brief Description:

3.3 (*3 points*): *Determine whether there are at least two (not necessarily consecutive) days that have the same value, i.e., does there exist* $i \neq j$ *such that* $A[i] = A[j]$.

Running Time:

Brief Description:

3.4 (*3 points*): *Suppose the entire sequence is initially increasing and then decreasing, i.e., for some unknown value* r*,*

$$A[1] < A[2] < \ldots < A[r] > A[r+1] > A[r+2] > \ldots > A[n] \ .$$

Compute the maximum value of the stock, i.e., $\max_{1 \le t \le n} A[t]$.

Running Time:

Brief Description:

Question 4. (*10 points*) Let $T = (V, E)$ be a balanced binary tree with $n = 2^k - 1$ nodes including a root named r. Suppose every node $v \in V$ has an associated weight $w(v)$ that can be either positive or negative. We say $T' = (V', E')$ is a *rooted subtree* of T if

1. $V' \subseteq V$ and $E' \subseteq E$.
2. $r \in V'$ and whenever u is a node in T' then the parent of u is also in V'.
3. Whenever $u, v \in V'$ and (u, v) in an edge of T then (u, v) is an edge of T'.

We say the weight of T' is $\sum_{v \in V'} w(v)$.

4.1 (*2 points*): *If $n = 3$, how many rooted subtrees does T have?*

4.2 (*2 points*): *Solve the recurrence $f(n) \leq 2f(n/2)+c$ and $f(1) = c$ where c is a positive constant.*

4.3 (*6 points*): *Design and analyze an efficient algorithm for finding the rooted subtree of maximum weight. Remember to explain why your algorithm is correct and give the running time.*

Question 5. (*10 points*) Suppose you have n assignments due in the next 24 hours. You are going to start immediately and not stop until you have finished them all. It is up to you in which *order* you do the assignments. Suppose the ith assignment takes t_i time and has a deadline at d_i.

5.1 (*2 points*): *As an example, suppose you have three assignments and* $t_1 = 5, t_2 = 4$, *and* $t_3 = 4$. *Is it possible to complete the assignments before their deadlines if their deadlines are:*

$d_1 = 8, d_2 = 5,$ *and* $d_3 = 13$*?* **Yes** **No**

$d_1 = 9, d_2 = 4,$ *and* $d_3 = 13$*?* **Yes** **No**

5.2 (*2 points*): *Suppose there is at least one ordering in which all assignments are completed before their deadlines. Prove that if you complete the assignments in order of increasing deadline then all assignments will be completed before their deadline.*

5.3 (*2 points*): *Suppose the* i*th assignment takes* $2i$ *minutes to complete and has a deadline after* $i(i+1)$ *minutes. Can all the assignments be completed before their deadlines? Justify your answer.*

5.4 (*4 points*): *Suppose there exists an assignment, such that if you skip this assignment (i.e., spend no time on it), you can complete all the other assignments before their deadlines. Consider the following part of an outline of algorithm for finding such an assignment:*

1. Sort assignments such that the ith assignment has the ith smallest deadline.
2. For all i, compute $f(i) = t_1 + \ldots + t_i$, $\ell(i) = f(i) - d_i$, and $m(i) = \max(\ell(i), \ell(i+1), \ldots, \ell(n))$.
3. ???

How fast can step 1 be performed?

How can step 2 be performed in $O(n)$ time?

Suggest a step 3 that returns the answer and can be performed in $O(n)$ time. Justify your answer.