ICS/CSE 46 Final Exam

Read all of the following information before starting the exam:

• Show all work, clearly and in order, if you want to get full credit.
• Circle or otherwise indicate your final answers.
• Please keep your written answers brief; be clear and to the point. I may take points off for rambling and for incorrect or irrelevant statements.
• This test has 16 problems on 8 total pages, including this one, the score sheet page, and a scratch page. It is your responsibility to make sure that you have all of the pages!
• Each problem is worth 10 points for a total of 160 points on this test.
• If you are going to write a portion of an answer on the back of a page, clearly indicate you are doing so on the front of that same page.
• Good luck!
## Score sheet

<table>
<thead>
<tr>
<th>Problem</th>
<th>Score</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>
1. Please clearly mark the following statements as True or False. Do not write a single letter; spell the words out.

(a) The Bellman-Ford algorithm does not work correctly when there are cycles with negative total weight.  

(b) The lightest weight edge will always be in the minimum spanning tree of a graph.  

(c) DAGs have \( n \) strongly connected components.  

(d) NP stand for “Not Polynomial”.  

(e) Dijkstra’s algorithm performs at most \( n \) update operations and \( m \) remove minimum operations on a priority queue.  

2. For each pair of functions below circle the true statements.

(a) \( f(n) = \frac{n}{3} \) and \( g(n) = n^{1/2} \)

\[ f(n) = \Omega(g(n)) \quad f(n) = \Theta(g(n)) \quad f(n) = O(g(n)) \]

(b) \( f(n) = 10n \log n + 5n \) and \( g(n) = n \log n - n \)

\[ f(n) = \Omega(g(n)) \quad f(n) = \Theta(g(n)) \quad f(n) = O(g(n)) \]

(c) \( f(n) = n \) and \( g(n) = 2^n \)

\[ f(n) = \Omega(g(n)) \quad f(n) = \Theta(g(n)) \quad f(n) = O(g(n)) \]
3. (a) Write down all of the lists that Mergesort will make recursive calls on when being run on the list 4,3,6,1,2,7,5,8.

(b) List all of the swaps that take place when Insertion sort is run on the list 4,1,2,3,6,5,8,7.

4. (a) Suppose a hash table is storing numbers and the hash function it uses is to sum the digits of the numbers. Give THREE numbers who will all collide no matter the size of the hash table.

(b) What bad behavior can happen if a poor hash function is chosen for use in a hash table?
5. Draw the directed graph corresponding to this adjacency list:

\[
\begin{array}{c|ccccc}
A & B & C & E \\
B & A & & & \\
C & C & D & & \\
D & A & & & \\
E & D & & & \\
\end{array}
\]

6. Fill in the adjacency matrix for the following directed graph:

\[
\begin{array}{c|ccccc}
1 & 2 & 3 & 4 & 5 \\
1 & & & & & \\
2 & & & & & \\
3 & & & & & \\
4 & & & & & \\
5 & & & & & \\
\end{array}
\]
7. Draw a binary rooted tree that is AVL balanced, but with height larger than the minimum possible height for that number of nodes.

8. (a) Heavily thicken the edges of the minimum spanning tree in the following graph.

(b) When using Kruskal’s algorithm on the above graph, list the edges of the minimum spanning tree in the order they would be added.
9. In the above graph, run Dijkstra's algorithm from vertex A and write in order the distances assigned to vertex C.

10. (a) Draw a graph with at least one negative edge weight and identify a start vertex, so that running Dijkstra's algorithm from that start vertex will NOT compute the correct distances.

(b) Draw a graph with at least one negative edge weight and identify a start vertex, so that running Dijkstra's algorithm from that start vertex still computes the correct distances.
11.  (a) Draw the above binary heap after inserting 14.

(b) Draw the original binary heap (not the one with 14) after executing a remove minimum operation.
12. (a) Draw the resulting heap from running the linear time build heap method on the above list.

(b) Draw the resulting heap from inserting each element of the above list in order into an empty heap.
13. (a) Write down a sequence of union find operations that will create an element that has a rank of 3.

(b) Write a sequence of find operations that will follow the previous parts and compress all the paths as short as possible.
14. In this class, we discussed basic binary search trees and a self balancing binary search tree (AVL trees). There exists many other types of binary search trees that we did not discuss. What is the fastest possible time to insert an element into any binary search tree and why? Hint: How fast can you sort using a binary search tree?
15. Describe an algorithm for checking if a given directed graph is a DAG.
16. A graph is two colorable if using two colors you can assign each vertex one of the colors such that all pairs of neighboring vertices have different colors. For example, below on the left is a two coloring of a graph; the black vertices are only adjacent to the white vertices. On the right is a graph that cannot be two colored, in the example an attempted two coloring has two adjacent black vertices. Describe an algorithm to check if a graph is two colorable.
Scratch Paper