Mid-Term Exam 1

COMP 251 Algorithms and Data Structures
Tues. Feb. 4, 2014
Prof. Michael Langer

STUDENT NAME: _____ ID: _____

Instructions:

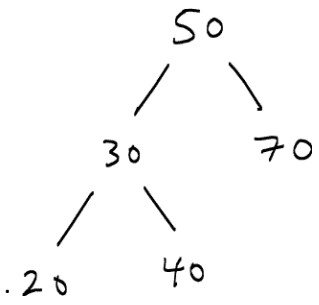
- You may use a small number of pages of notes.
- No electronic devices are allowed.
- If your answer does not fit on a page, then use the reverse side.

.

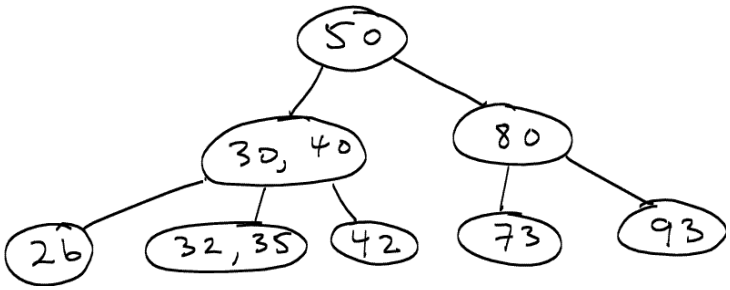
question	points	score
1	4	
2	4	
3	4	
4	4	
5	6	
6	4	
7	2	
8	2	
TOTAL	30	

1. (4 points)

a) Insert the key **45** into the AVL tree shown below and balance the tree if necessary.



b) Insert the key **38** into the following 2-3 tree and balance the tree if necessary.



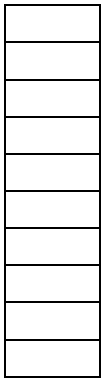
2. (4 points)

Consider the following sequence of (key, value) pairs, which defines a map, and a hashcode for each key. Assume these entries are **put** into a hash table with m=10 slots.

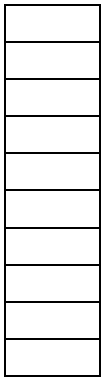
("C", "cal") 79
("D", "doug") 53
("B", "bob") 88
("A", "abe") 59
("F", "fred") 71
("E", "ed") 8
("G", "greg") 20

Draw the resulting hash tables for the case of:

a) open hashing (using a linked list in each bucket to handle collisions)



b) closed hashing (using linear probing to handle collisions)



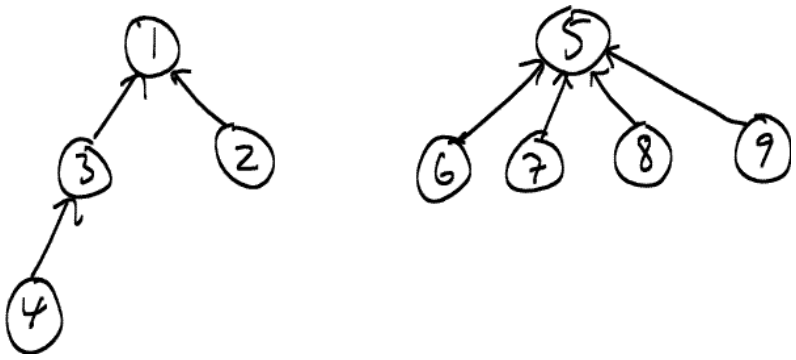
3. (4 points)

a) Given an input list of six keys [9, 6, 5, 4, 3, 8] which are placed into an array in the given order, build a min-heap using the O(n log n) algorithm that is based on upheap. You may draw the heap as a tree rather than an array.

b) The same question as (a) but now build a heap using the O(n) algorithm that is based on downheap.

4. (4 points)

Given the following forest which was constructed from a sequence of union operations,



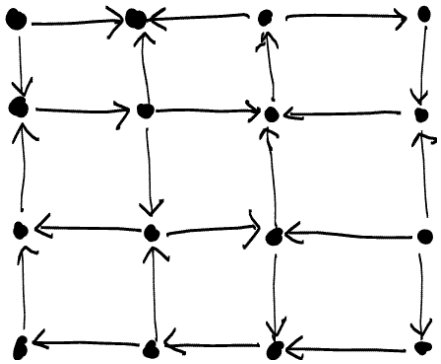
draw the result of applying **union(3,6)** followed by **union(2,4)**, using:

a) "union-by-size" with path compression

b) "union-by-height" with path compression.

5. (6 points)

a) Circle *all* of the strongly connected components of the following graph.

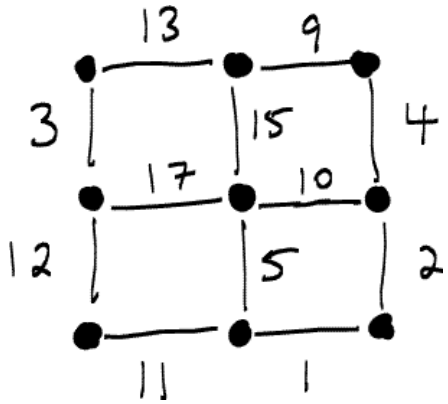


b) How many strongly connected components does a directed acyclic graph (DAG) of n vertices have? Possible answers are 0, 1, n, or "it depends". Circle one of these answers *and fully justify it*. You will receive no points for just getting the correct answer.

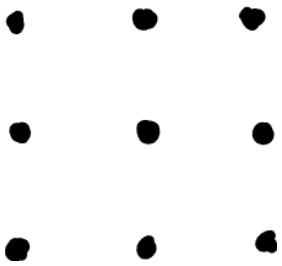
c) Prove that if there is a path from vertex v to vertex w in a DAG, then vertex v must come before w in any topological ordering.

6. **(4 points)**

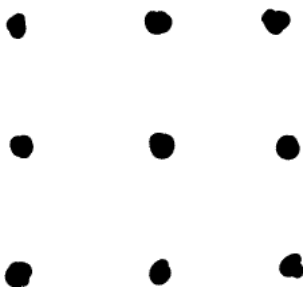
Consider the following graph.



a) Starting at the source vertex in the upper left corner, draw the tree that is found by Dijkstra's algorithm for single source shortest paths. Label the vertices (as you go) by the lengths of the shortest paths.



b) Draw the minimal spanning tree.



7. (2 points)

We saw in assignment 1 how to use a hashmap (nameToIndex) to index into a priority queue and how this could allow us to changePriority in O(log n) time. Would there be any advantage or disadvantage *in terms of O( ) time* if we used a balanced binary search tree for this map, instead of a hashmap? Briefly explain.

Hint: Binary search trees can be used to organize keys in a map, if the keys are comparable e.g. Java's TreeMap which uses a red-black tree.

8. (2 points)

Kruskal's algorithm for computing the minimal spanning tree begins by sorting all the edges by their cost. Sorting takes time O(|E| log |E|) and, as discussed in class, this step is the bottleneck of the algorithm. Suppose Kruskal's algorithm used a priority queue of edges instead of sorting the edges. Would this give *overall* faster, same, or slower performance in the O( ) sense? Justify your answer.