CSC 225 A01 FALL 2012 (CRN: 10414)
ALGORITHMS AND DATA STRUCTURES I
FINAL EXAMINATION
UNIVERSITY OF VICTORIA

1. Student ID: ______________________
2. Name: __________________________
3. DATE: 7 DECEMBER 2012
   DURATION: 3 HOURS
   INSTRUCTOR: V. SRINIVASAN
4. THIS QUESTION PAPER HAS FOURTEEN PAGES (INCLUDING THE COVER PAGE).
5. THIS QUESTION PAPER HAS EIGHT QUESTIONS.
6. ALL ANSWERS TO BE WRITTEN ON THIS EXAMINATION PAPER.
7. THIS IS A CLOSED BOOK EXAM. NO AIDS ARE ALLOWED.
8. KEEP YOUR ANSWERS SHORT AND PRECISE.

Q1 (10)	
Q2 (10)	
Q3 (10)	
Q4 (10)	
Q5 (10)	
Q6 (10)	
Q7 (10)	
Q8 (10)	
TOTAL (80) =	

1. **Asymptotic Analysis.**

(i) Show that $f(n) = 4 \log n + \log \log n$ is $\Theta(\log n)$. [5 Marks]

(ii) Show that $f(n) = 12n^2 + 6n$ is $o(n^3)$ and $\omega(n)$. [5 Marks]

2. **Solving Recurrence Equations.**

Solve the following recurrence equation to get a closed-formula for $T(n)$. Assume the n is a power of two. [10 Marks]

$$\begin{aligned} T(n) &= 1 \text{ if } n = 1 \\ &= T\left(\frac{n}{2}\right) + n \text{ if } n \geq 2 \end{aligned}$$

3. **Sorting Algorithms**.

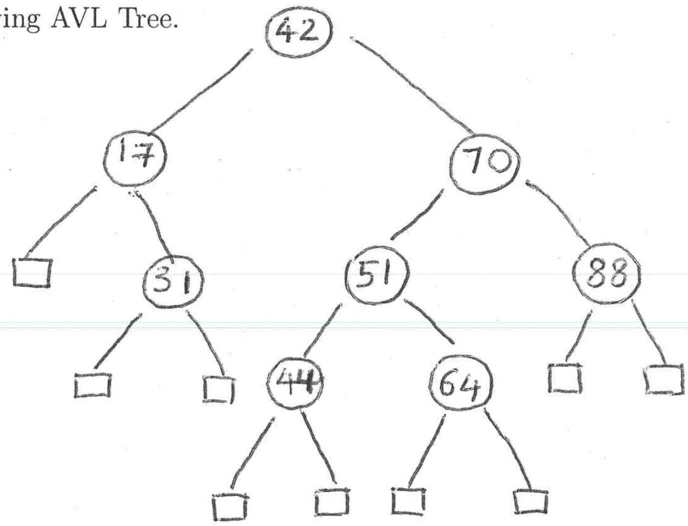
(i) The priority queue ADT has many possible implementations, each with its own running time for the two operations, insertItem and removeMin. Each such implementation is closely related to a sorting algorithm. Complete the following table assuming an input of size n. [6 Marks].

Priority Queue Implementation	*Time for insertItem*	*Time for removeMin*
Unsorted Array		
Sorted Array		
Heaps		

(ii) Show how Radix sort runs on the input: 12 62 51 25 47 58 92 64 70 19. [4 Marks]

4. **Balanced Search Trees.**

Consider the following AVL Tree.

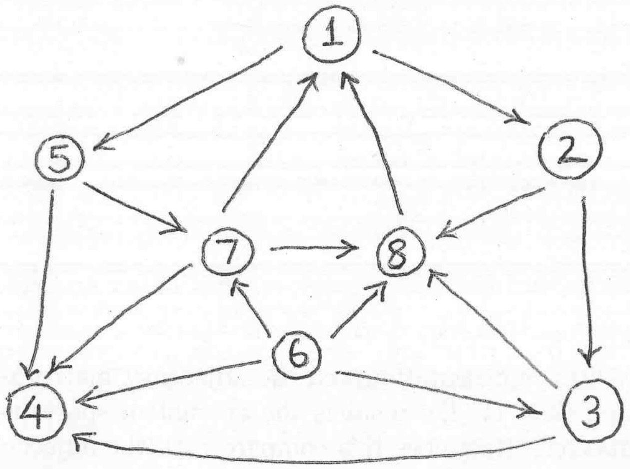


(i) Draw the resulting AVL tree after the insertion of key 62. Show all the steps of the restructuring procedure, if any. [6 Marks]

(ii) Show using an example how more than one restructure operation could be necessary after a deletion from an AVL tree. [4 Marks]

5. **Graph Traversal.**

Consider the following directed graph G.

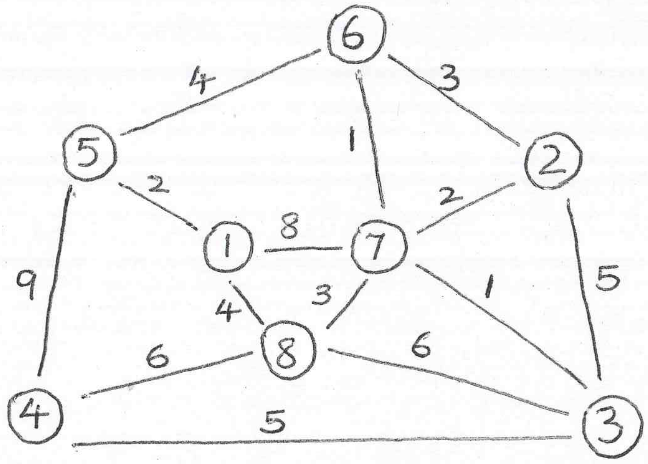


(i) Perform a DFS traversal on G starting at vertex 1 and compute the preorder and postorder listing of the vertices of G. Assume that, in the traversal, the adjacent vertices are visited in the increasing order of vertex labels. [5 Marks]

(ii) Draw the adjacency-lists representation and the adjacency matrix representation of G. For any graph $G = (V, E)$, what is the amount of space used by the adjacency-lists representation? How does this compare with the adjacency matrix representation? [5 Marks]

6. **Minimum Spanning Trees.**

Consider the following undirected, weighted graph G.



(i) Find the minimum spanning tree of G using Kruskal's algorithm. Clearly list all the edges in the order in which they are discovered. What is the weight of the MST? [6 Marks]

(ii) Prove the following property for minimum spanning trees, known as cycle property: Given any cycle in an edge weighted graph (all edge weights distinct), the edge of maximum weight in the cycle does not belong to the MST of the graph. [4 Marks]

7. **Shortest Paths.**

Consider the undirected, weighted graph G given in Problem 6.

(i) Find a shortest path from the source vertex 1 to every vertex of G using Dijkstra's algorithm. Clearly indicate the vertex chosen at each step and the updated distance values maintained by the algorithm at the end of each step. [6 Marks]

(ii) Give an example of a graph to show why Dijkstra's algorithm does not compute shortest paths correctly if the graph has edges with negative weights. Clearly indicate the source vertex and the vertex for which the algorithm makes an error. [4 Marks]

8. **Lower Bound Techniques.**

In this question, we will prove a lower bound on the running time of any comparison-based algorithm for the following search problem on a sorted array: Given a sorted array $A[0, \ldots, n-1]$ and an element x, find an index i, $0 \leq i \leq n-1$, such that $A[i] = x$. Else, output "not found".

(i) Describe, using **pseudo-code**, how binary search algorithm solves this problem. Represent the algorithm using a decision tree for an input array A of size 7. [6 Marks]

(ii) Show that any decision tree for a comparison-based algorithm for the search problem on a sorted array must have at least $n + 1$ leaves and hence it must take $\Omega(\log n)$ time. [4 Marks]

END OF EXAM