CSC 225 A01/A02 SPRING 2013 (CRN: 22634 and 20294) ALGORITHMS AND DATA STRUCTURES I FINAL EXAMINATION UNIVERSITY OF VICTORIA

1. Student ID: ____________________
2. Name: ______________________
3. DATE: 15 APRIL 2013
   DURATION: 3 HOURS
   INSTRUCTOR: V. SRINIVASAN
4. THIS QUESTION PAPER HAS TEN 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 CALCULATORS AND OTHER AIDS ARE ALLOWED.
8. READ THROUGH ALL THE QUESTIONS AND ANSWER THE EASY QUESTIONS FIRST. 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. **[Sorting Algorithms]**

   (i) Let $T(n)$ denote the running time of MergeSort on an input of size n. What is the recurrence equation for $T(n)$? [3 Marks]

   (ii) Solve this recurrence equation to obtain the running time for MergeSort on an input of size n. What is its running time? [7 Marks]

2. **[Lower Bound for Sorting]**

   Consider a comparison based sorting algorithm for sorting an input of n numbers $x_1x_2 \ldots x_n$ where n is even. Suppose that the sorting algorithm is also given the following additional information: $x_1, x_3, \ldots, x_{n-1}$ will be in the first half of the sorted order and $x_2, x_4, \ldots, x_n$ will be in the second half of the sorted order. Show that any comparison based sorting algorithm still requires $\Omega(n \log n)$ time to sort $x_1x_2 \ldots x_n$ even if it is given this additional information. [10 Marks] (Hint: what is the number of leaves in the decision tree now?)

3. **[Balanced Search Trees]**

(i) What are the two properties that any AVL tree must satisfy? [6 Marks]

(ii) What is the height of any AVL tree T on n nodes? Give a proof. [4 Marks]

4. **[Hashing]**

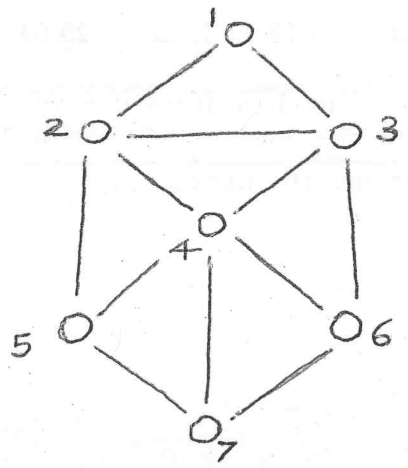
   Suppose that a hashing scheme stores a set of 10 keys

   $$90\ 18\ 36\ 12\ 41\ 10\ 28\ 38\ 25\ 54$$

   using the hash function $h(k) = k \bmod 13$. It resolves collisions using chaining. Draw the hash table with all the 10 keys inserted. Calculate the load factor of the hash table at the end of the insertions. [10 Marks]

5. **[Graph Representation]**

Consider the following undirected graph G.

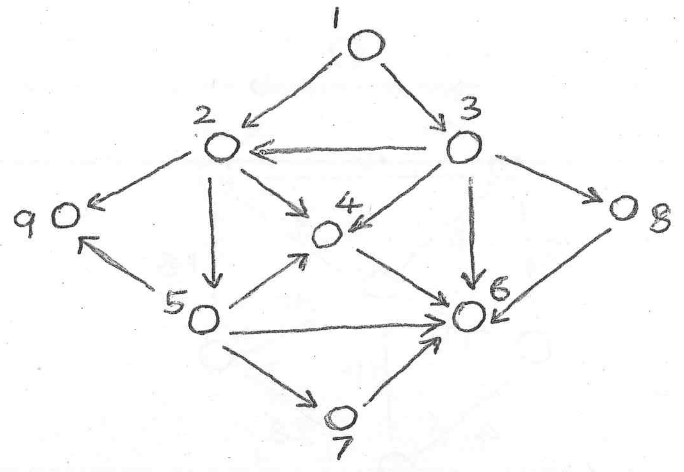


(i) Draw the adjacency-lists representation and the adjacency matrix representation of G. [8 Marks]

(ii) Give an example of a graph operation for which adjacency lists are better than adjacency matrices and one operation for which adjacency matrices are better than adjacency lists. [2 Marks]

6. **[Graph Traversal]**

Consider the following directed graph G.

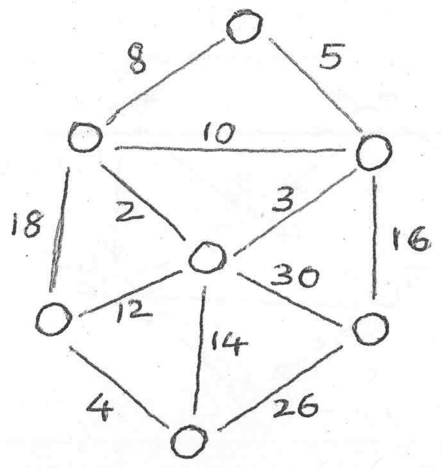


(i) Perform a DFS traversal on G starting at vertex 1 and compute the pre-order and post-order listing of the vertices of G. Assume that, in the traversal, the adjacent vertices are visited in the increasing order of vertex labels. [8 Marks]

(ii) For any connected, undirected graph G, what is the order in which you can delete the vertices of G so that G remains connected at every stage? [2 Marks]

7. **[Minimum Spanning Trees]**

Consider the following undirected, weighted graph G.



(i) Find the minimum spanning tree (MST) of G. Clearly list all the edges in the order in which they are discovered by your algorithm. What is the weight of the MST? [7 Marks]

(ii) Prove that a MST of a graph G is unique if all the edge weights in G are distinct. [3 Marks] (Hint: Try proof by contradiction - Start with the assumption that the graph has two distinct MSTs and show that this leads to a contradiction)

8. **[Shortest Paths]**

(i) Given an example of a graph to show why Dijkstra's algorithm does not compute single source 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. [5 Marks]

(ii) Here is a claim by Bob to make Dijkstra's algorithm work for computing single source shortest path even in graphs with negative edge weights: Add a large enough constant to every edge weight, so that all of them are positive and then run Dijkstra's algorithm. This will not change the solution to single source shortest path. Is Bob correct? If not, give a counterexample. [5 Marks]

END OF EXAM