CSC 225 A01 SPRING 2016 (CRN: 20717)
ALGORITHMS AND DATA STRUCTURES I
FINAL EXAMINATION
UNIVERSITY OF VICTORIA

1. Student ID: ______________________

2. Name: ______________________

3. DATE: 11 APRIL 2016
   DURATION: 3 HOURS
   INSTRUCTOR: V. SRINIVASAN

4. THIS QUESTION PAPER HAS 14 PAGES INCLUDING THE COVER PAGE.

5. THIS QUESTION PAPER HAS TEN QUESTIONS.

6. ALL ANSWERS TO BE WRITTEN ON THIS EXAMINATION PAPER. USE THE LAST TW0 BLANK PAGES FOR CALCULATIONS.

7. THIS IS A CLOSED BOOK, CLOSED NOTES EXAM. NO CALCULATORS ARE ALLOWED.

8. READ THROUGH ALL THE QUESTIONS AND ANSWER THE EASY QUESTIONS FIRST. **KEEP YOUR ANSWERS SHORT AND PRECISE.**

Q1 (5)	
Q2 (5)	
Q3 (5)	
Q4 (5)	
Q5 (5)	
Q6 (5)	
Q7 (5)	
Q8 (5)	
Q9 (5)	
Q10 (5)	
TOTAL (50) =	

1. **Asymptotic Notation**

   (a) Order the following functions by big-Oh notation. All logarithms are to base 2. [3 Marks]

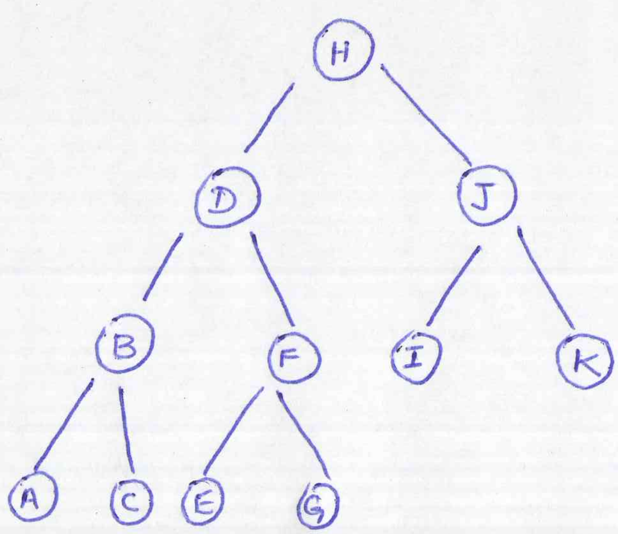
$$n!, 2^{\log n}, \sqrt{n}, 1000, \sqrt{\log n}, n^2 \log n, 5 \log \log n, 2^n, n^{\log n}, n^{3/2}$$

   (b) Prove that 2^{n+1} is $O(2^n)$. Prove that 2^{2n} is not $O(2^n)$. [2 marks]

2. **Basic Data Structures**

(a) What are the minimum and the maximum number of nodes in any heap of height h? [1 Mark]

(a) For the binary tree below, output the sequence of nodes in the order in which they are visited in the preorder and inorder traversal. [2 Marks]



Traversal	Sequence
Preorder	
Inorder	

(b) Describe the main steps of an algorithm (in clear pseudo-code) that constructs and outputs a binary tree, given its preorder and inorder traversal sequences as input. [2 Marks]

3. **Sorting**

Show how Quick-Sort algorithm works on the following input sequence S using the quick-sort tree.

Use the following pivot rule that picks the element in the "middle" - For an array $A[0, 1, \ldots, n-1]$ of size n, it uses the element in $A[n/2]$ as pivot if n is even and the element in $A[(n-1)/2]$ as pivot if n is odd [3 Marks].

$$S = [65\ 24\ 53\ 45\ 18\ 32\ 86\ 47]$$

(b) You are given a set of n distinct keys. After a preprocessing algorithm you should be able to compute queries of the following form in $\mathbf{O(1)}$ time:
Input: Two integers i and j, $1 \leq i \leq j \leq n$
Output: The sum of the ith largest, plus the $(i+1)$st largest, plus the $(i+2)$nd largest, up until the jth largest.

Highlight the main steps of the preprocessing algorithm that you need. What is the total running time of your preprocessing algorithm? [2 Marks]

4. **[The Master Theorem]**

$$\begin{aligned} T(n) &= c \text{ if } n < d \\ &= aT\left(\frac{n}{b}\right) + f(n) \text{ if } n \geq d \end{aligned}$$

(a) If there is a small constant $\epsilon > 0$ such that $f(n)$ is $O(n^{\log_b a - \epsilon})$, then $T(n)$ is $\Theta(n^{\log_b a})$.

(b) If there is a small constant $k \geq 0$ such that $f(n)$ is $\Theta(n^{\log_b a} \log^k n)$, then $T(n)$ is $\Theta(n^{\log_b a} \log^{k+1} n)$.

(c) If there are small constants $\epsilon > 0$ and $\delta < 1$ such that $f(n)$ is $\Omega(n^{\log_b a + \epsilon})$ and $af(n/b) \leq \delta f(n)$, then $T(n)$ is $\Theta(f(n))$.

Solve the following recurrence equations using the Master Theorem given above (assuming that $T(n) = c$ for $n < d$, for constants $c > 0$ and $d \geq 1$). [5 Marks]

(a) $T(n) = 8T(n/2) + n^2$

(b) $T(n) = 27T(n/3) + n^3 \log n$

(c) $T(n) = 4T(n/2) + n^3$

5. **Binary Search Trees**

(a) Insert items with the following keys (in the given order) into an initially empty binary search tree:

$$30, 40, 24, 58, 48, 26, 11, 13$$

Draw the tree after each insertion. [3 Marks]

(b) Delete the item with key 24 from the binary search tree obtained at the end of all the insertions in Part (i). Draw the tree after the deletion. [1 Mark]

(c) For each of the following operations supported by a binary search tree, write down its worst-case running time for an input of size n. [1 Mark]

Operation	Worst-Case
findItem	
deleteItem	
insertItem	

6. **Hashing**

(a) Suppose that a hashing scheme stores a set of 8 keys

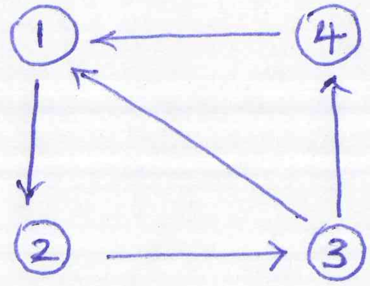
$$5 \quad 11 \quad 19 \quad 15 \quad 20 \quad 33 \quad 12 \quad 17$$

using the hash function $h(k) = (2k+1) \bmod 9$. It uses open addressing and resolves collisions using linear probing. Draw the hash table with all the 8 keys inserted. [4 Marks]

(b) Give a general expression/formula for all keys that are mapped to slot 2 by h. Note that the slots of the hash table are indexed from 0 to 8. [1 Mark]

7. **Graph Representation**

Consider the following directed graph G.

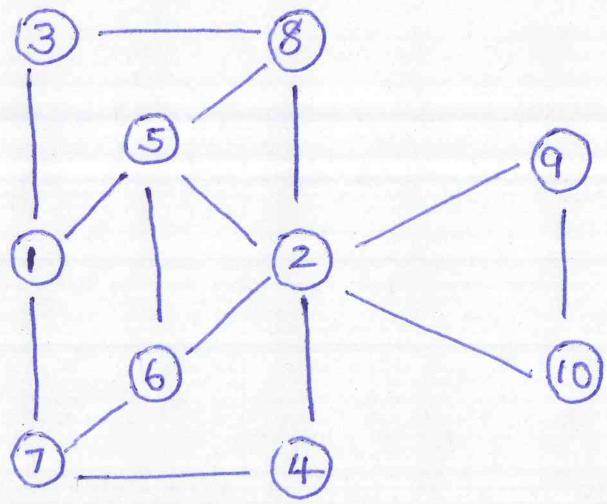


(a) Give the adjacency list and adjacency matrix representation of G. [3 Marks]

(b) First, compute $G^2 = G \times G$ using matrix multiplication. Using the result, write down the adjacency matrix of G^2. Recall that G^2 is a graph that has the same set of vertices as G and contains an edge between two vertices if there is a path of length 2 between them in G. [2 Marks]

8. **Graph Traversal I**

Consider the following undirected graph G.



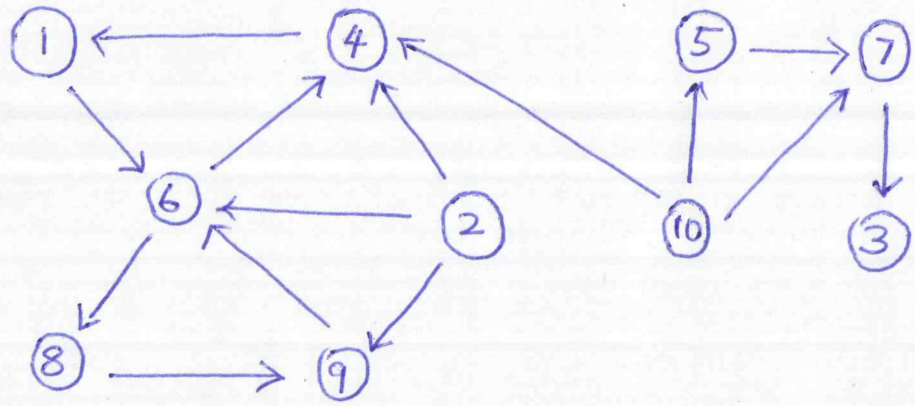
(a) Perform a BFS traversal on G starting at vertex 1. Assume that, in the traversal, the adjacent vertices are visited in the increasing order of vertex labels. List the BFS tree edges in the same order in which they are found. [3 Marks]

(b) Output the first edge during the BFS traversal that indicates the presence of an even cycle in G. [1 Mark]

(c) Output the first edge during the BFS traversal that indicates that G is not bipartite. [1 Mark]

9. **Graph Traversal II**

Consider the following directed graph G.



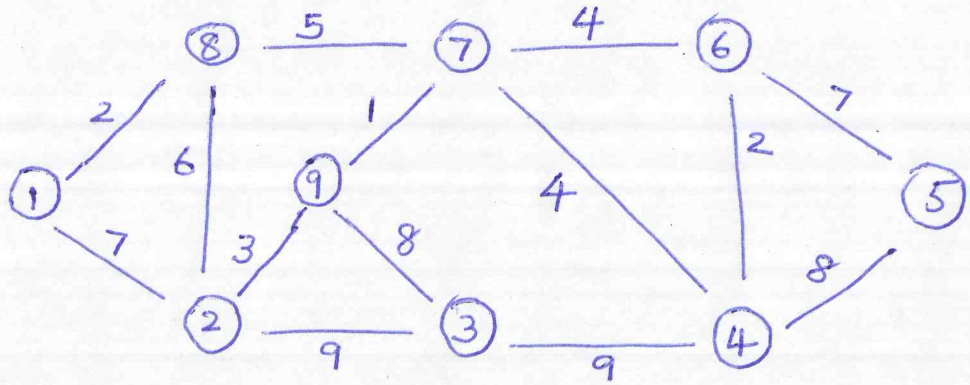
(a) Perform a DFS traversal on G starting at vertex 1. **The traversal must visit all the vertices of** G. Assume that, in the traversal, the adjacent vertices are visited in the increasing order of vertex labels. Compute the preorder and postorder listing of the vertices. [3 Marks]

Traversal	Sequence
Preorder	
Postorder	

(b) Using the postorder listing obtained Part (a) and Kosaraju's algorithm, compute the strongly connected components of G. You have to list the strongly connected components in the order in which they are output by the algorithm. [2 Marks]

10. **Minimum Spanning Trees.**

Consider the following undirected, weighted graph G.



(a) Compute the minimum spanning tree (MST) of G. Clearly list all the edges in the order in which they are discovered by your algorithm. In the table above, an edge label is the pair of end points of that edge. [3 Marks]

Edge Label	Weight
Total Weight	

(b) Describe the cut property that is used to give a proof of correctness of all the algorithms for computing Minimum Spanning Trees. [2 Marks]

BLANK PAGE FOR CALCULATIONS

BLANK PAGE FOR CALCULATIONS

END OF EXAM