Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY SRINAGAR

Course: Data Structures — **IVth Semester:** CSE

Exam: MAJOR Spring 2019 — **Max Marks:** 60 — **Time:** 3 hrs

CO1	Ability to understand basic data structures and Abstract Data Types along with their definitions and usage in computer science.
CO2	Understanding and implementing linear data structures. Ability to implement both static and dynamic linear data structures like linked lists, stacks and queues. Implementing and understanding the usage of non linear data structures like trees.
CO3	Applications of data structures with some working examples in stacks, queues, linked lists and trees. Memory and storage management concepts. Hashing.
CO4	Graph representations and algorithms to work on graphs. Sorting, searching algorithms and their implementations using basic data structures.

Attempt any four questions from 1-5

a. Answer any five briefly [5*3 = 15] (CO2, CO3,CO3,CO3,CO2,CO4):

a. Evaluate **6 5 2 3 + 8 * + 3 + * 4 /** using stacks.

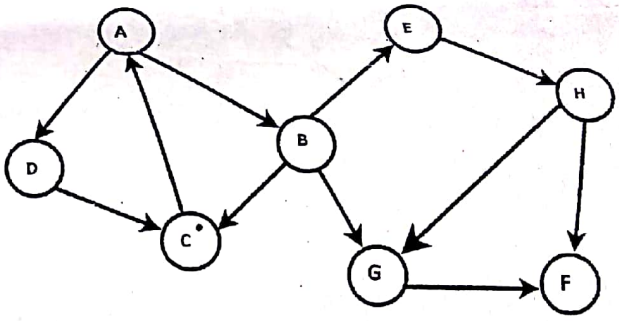
b. Explain Internal and External Fragmentation for Variable Block Allocation Technique.

c. What is an AVL tree? Explain how AVL tree maintains its properties?

d. Prove mathematically that the height of a complete binary search tree is **log n** where **n** is the number of nodes of the binary search tree.

e. Write a function that takes the root of a doubly linked list, counts the number of elements and prints the list in reverse.

f. Consider the following graph. If there is ever a decision between multiple neighbour nodes in the BFS or DFS algorithms, assume we always choose the letter closest to the beginning of the alphabet first. Write down the BFS and DFS sequence with A as the starting point.



2. Answer the following[6,6,3] (CO3,CO2/CO3,CO3):

a. **Construct** a binary tree T from the given pre-order and in-order traversal sequences, and answer the questions that follow.

Pre-order: A,B,E,H,Q,R,C,D,K,L,M

In-order: B,Q,R,H,E,A,D,L,K,M,C

i. What is the maximum height of the tree if root node is at height 1?

ii. Write down the internal and leaf nodes of the binary tree.

iii. What is the post-order traversal sequence of the binary tree?

b. **Using stacks**, convert the following expression into an expression tree. No marks will be awarded for direct conversion (without steps).

A + ((B * D + E / F) ^ G – H) / (C + E * F) – B * E

c. Write a function that takes the root of a binary search tree and returns the number of leaf nodes in the tree.

3. **Answer the following[8,7] (CO3, CO1/CO3):**

a. Show the result of inserting **15, 17 , 6, 19, 11, 10, 13, 20, 8, 14, 12, 5** one at a time (in sequence) into an initially empty binary min heap. Show the result of performing ***three subsequent delete min*** operations in the final binary min heap obtained.

b. Given **input {4371, 1323, 6173, 4199, 4344, 9679, 1989}** and a **hash function h(x) =x mod 10.** Prepare the resulting hash table for the following:

i. Separate chaining hash table.

ii. Open addressing hash table using linear probing.

iii. Open addressing hash table using quadratic probing.

iv. Open addressing hash table with **second hash $h_2(x) = 7 - (x \bmod 7)$.**

4. **Answer the following[3,6,6] (CO4, CO4,CO1/CO4):**

a. What is the output of quick sort **after the 3rd iteration** given the following sequence? Complete the sorting procedure given the information below:

Initial Sequence: 24, 56, 47, 35, 10, 90, 82, 31

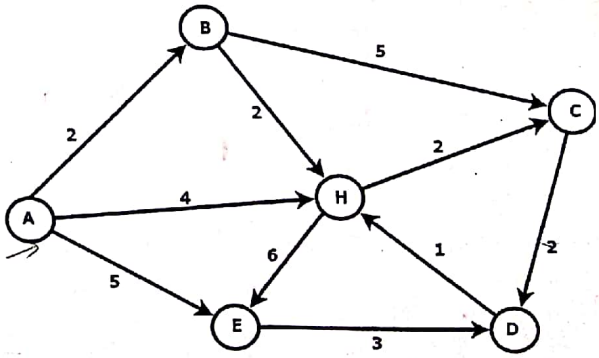
Pass 1:- (10) 24 (56 47 35 90 82 31)
Pass 2:- 10 24 (56 47 35 90 82 31)
Pass 3:- 10 24 (47 35 31) 56 (90 82)

b. For the array in **part (a),** apply **Insertion and Selection sort** to sort the array in **ascending order**. Show the two sorting procedures side by side in two columns depicting the changes in each pass of each sorting procedure.

c. Write a function that bubble sorts elements of an array entered by the user at runtime. The function should print the sorted sequence and stop once the array is sorted without additional comparisons.

5. **Answer the following: [7, 8] (CO4, CO4):**

a. Use Djikstra's Algorithm for finding the shortest path from source '**A**' to all other nodes in the graph. **Show all steps** of the algorithm and **shortest paths / parent nodes** as they change and arrive at the solution. Mention the shortest paths from '**A**' to each node in the end.



b. What is the difference between Prim's and Kruskal's algorithm for finding the Minimum Spanning Tree of a graph. Find out the minimum spanning tree for the graph (A-I) **using both algorithms.** Show the steps for both algorithms and the final tree.