B/L/TX

Reg. No:

VIT
Vellore Institute of Technology

Final Assessment Test – November 2024

Course: PMCA501L - Data structures and Algorithms

Class NBR(s): 3124/ 3187/3225 Slot: C2+TC2

Time: Three Hours Max. Marks: 100

- KEEPING MOBILE PHONE/ANY ELECTRONIC GADGETS, EVEN IN 'OFF' POSITION IS TREATED AS EXAM MALPRACTICE
- DON'T WRITE ANYTHING ON THE QUESTION PAPER

Answer ALL Questions

(10 X 10 = 100 Marks)

1. Consider the recurrence relation T(n)=2T(n/2)+n Use the Substitution Method, Recurrence Tree Method, to solve this recurrence and determine the time complexity of the algorithm. Compare the results obtained from each method.

2. Given the infix expression: A + B * (C - D) / E ^ F, convert it to postfix notation using a stack. Then, evaluate the postfix expression assuming the values of A=5, B=3, C=8, D=2, E=4, and F=2.

3. Explain how polynomials can be represented using linked lists. Provide a detailed description of the operations (such as addition and multiplication) on polynomials using linked lists, with examples and necessary algorithms.

4. Sort the following array using **Insertion Sort** and **Shell Sort** (using gap sequence 5, 3, 1):

   Array: [29, 10, 14, 37, 14, 3, 19, 7, 12, 42].

   Show the step-by-step process for both algorithms and compare their performance.

5. Insert the following keys into a hash table using **Open Addressing** with a hash function h(x)=x%7 and a table size of 7:

   Keys: [50, 700, 76, 85, 92, 73, 101]

   a) Use **Linear Probing** to resolve collisions and show the final hash table.

   b) Use **Quadratic Probing** to resolve collisions and show the final hash table. Compare the effectiveness of both methods.

6. Consider the following sequence of numbers: [40, 20, 60, 10, 30, 50, 70]. Insert these numbers into a **Binary Search Tree (BST)**. After constructing the BST, perform the following operations:

   a) Find the minimum and maximum values in the BST.

   b) Delete the node with the value 20 from the BST and show the resultant tree.

7. Perform both **Depth First Search (DFS)** and **Breadth First Search (BFS)** on the following graph, starting from vertex A:

V={A,B,C,D,E,F},E={(A,B),(A,C),(B,D),(C,E),(D,E),(E,F)}. Show the order in which the vertices are visited in both traversals and discuss the time complexity of DFS and BFS.

8. Using **Dijkstra's Algorithm**, find the shortest path from vertex A to all other vertices in the following weighted graph:

V={A,B,C,D,E},E={(A,B,2),(A,C,4),(B,C,1),(B,D,7),(C,E,3),(D,E,1)}. Show each step of the algorithm, including the distance updates and final shortest path tree.

9.a) Apply **Merge Sort** to the array:

[38, 27, 43, 3, 9, 82, 10].

Show each step of the recursive division and merging process. Analyze the time complexity of the algorithm based on the number of comparisons and recursive calls.

OR

9.b) Explain the **Huffman Coding** algorithm and its use in data compression. Given the following characters and their frequencies, construct the Huffman tree and determine the corresponding Huffman codes for each character: Characters: {A,B,C,D,E,F}

Frequencies: {5,9,12,13,16,45}

10.a) i) Discuss the importance of algorithms in computing. Illustrate with examples how the efficiency of an algorithm can impact the performance of a system. Explain different criteria used to evaluate algorithms. **[6]**

ii) Explain asymptotic notations with example. **[4]**

OR

10.b) Define the **Longest Common Subsequence (LCS)** problem and explain how dynamic programming is used to solve it. Given two sequences: X="ABCBDAB"X = "ABCBDAB", find the length of their longest common subsequence and show the steps involved in constructing the dynamic programming table. Provide the final LCS.

⇔⇔⇔B/L/TX⇔⇔⇔