Master of Computer Applications
MCAC 103: Data Structures
Unique Paper Code: 223421112 / 223421103
Semester-I
December-2024
Year of admission: 2023 and 2024

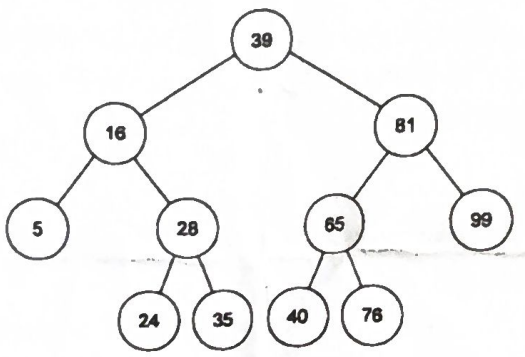
Time: Three Hours | Maximum marks: 70

Note: Answer all the questions. Your code/ algorithm/ pseudocode MUST be well documented and interfaces must be clearly defined.

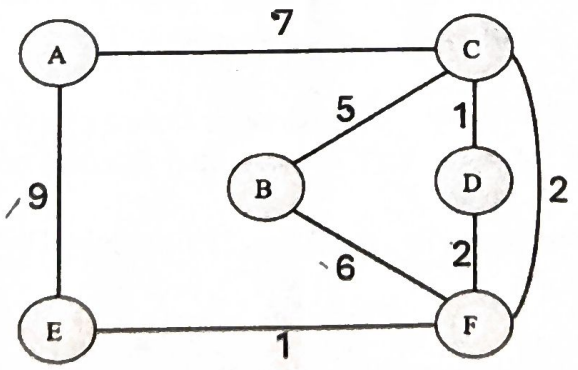
1. a. Determine the time complexity of the following code snippet. [4]

```
for (i = 1; i < n; i = i * 2) {
  for (j = i; j < n; j++)
    printf("x");
}
```

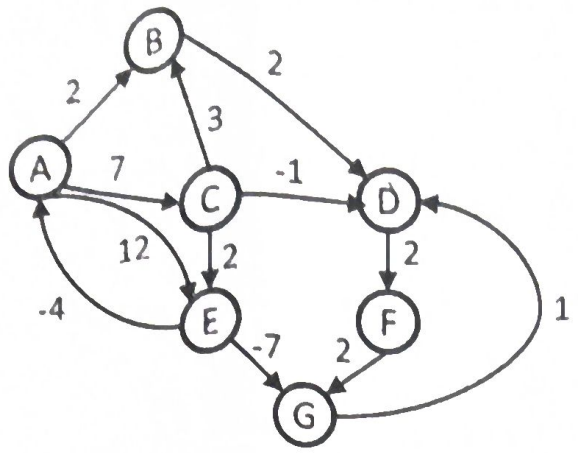
b. Show the resulting AVL tree after removing the node containing the value *81*. [4]



c. Show the step-by-step process of using Kruskal's algorithm with disjoint-set operations (Union-Find) to determine the Minimum Spanning Tree for the undirected graph given below. [6]



2. a. Consider the following directed, weighted graph: [7]



Despite the presence of negative weight edges in the graph, use Dijkstra's algorithm to compute the **supposedly shortest paths** from vertex A to all other vertices. Show your steps and also list the vertices in the order which you marked them known.

b. Consider a linear-probing hash table of size **10** with the hash function $h(x) = x \bmod 10$. After inserting six integer keys into an initially empty hash table, the array of keys is: [7]

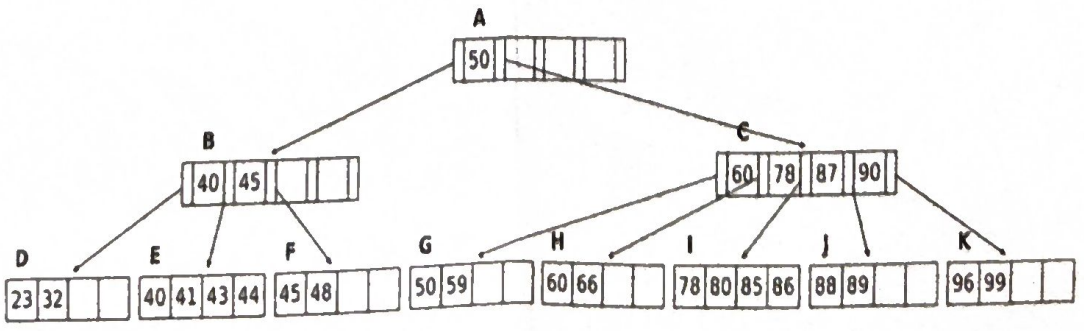
0	1	2	3	4	5	6	7	8	9
		42	23	34	52	46	33		

Which of the following choice(s) are insertion sequences resulting in the above hash table? Assume that the length of the hash table does not change during the insertions.

i. 46, 42, 34, 52, 23, 33
ii. 34, 42, 23, 52, 33, 46
iii. 46, 34, 42, 23, 52, 33
iv. 42, 46, 33, 23, 34, 52
v. 42, 23, 34, 52, 46, 33

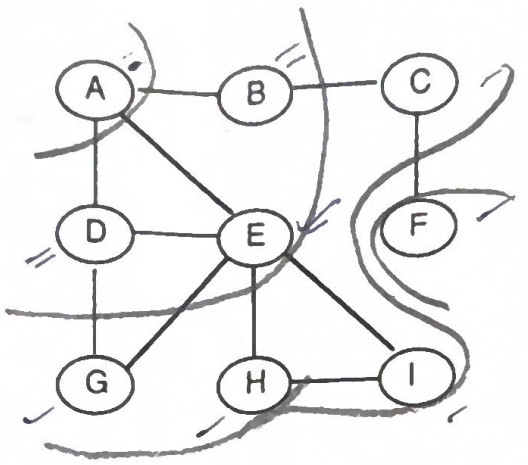
3. a. In a complete k-ary tree, every internal node has exactly k children or no child. [3] Find the number of leaves in such a tree with n internal nodes.

b. Consider the B+ tree shown below. Draw the updated B+ tree after inserting a data entry with key 79. [4]



c. Write the pseudo-code for a function that takes an array of integers as input and determines the nearest smaller element for each element in the array. For an element at index i, the nearest smaller element is the closest element to the left of i that is smaller than the element at i. If no such element exist, return -1 for that position. Your solution must use a stack, ensuring that the number of *push()* operations does not exceed the number of elements in the array. [7]

4. a. Given the graph in the figure below, specify the order in which the vertices will be traversed when the edges incident on a vertex are explored in alphabetical order of the adjacent vertices, starting from node A, for both *DFS* and *BFS* traversal algorithms. [6]



b. Give a clear description of an *efficient* algorithm for finding the k smallest elements of a very large n-element vector. Using $k=4$, show the steps of your algorithm on the following list of integers: 10, 9, 11, 14, 2, 15, 18, 7, 5, 21, 1. [8]

5. a. You are given a singly linked list where each node contains an integer value. Write an efficient algorithm to rearrange the list so that all odd-numbered elements (in terms of their values) appear before all even-numbered elements, maintaining the original relative order of odd and even elements. For example, given the linked list: $1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 6$, your algorithm should rearrange the list to: $1 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 2 \rightarrow 6$. Furthermore, the list should be scanned only once, and aside from a small amount of temporary space, no additional data structures should be used. [7]

b. Solve the following recurrence relation using the recursion tree approach. [7]

$$T(n) = 2T(n/2) + n \log n$$