Course Code : CST 252 / CDT 201 **EZFY/RW – 22 / 1229**

Third Semester B. Tech. (Computer Science and Engineering / Data Science) Examination

DATA STRUCTURE AND ALGORITHMS

Time : 3 Hours] [Max. Marks : 60

Instructions to Candidates :—

(1) Attempt all questions.

(2) All questions carry marks as indicated against them.

(3) Due credit will be given to neatness and adequate dimensions.

1. (A) Write an algorithm to order a list on N numbers in ascending sequence using an insertion sort. Compute the complexity of your algorithm using tabular method. State assumptions clearly. 5(CO1)

(B) Consider $6 \times 5 \times 7$ array, **A** which stores 64-bit hash values. The element A[2[[3][4] is stored at an address location 2768 in row major order.

Calculate the base address of A and compute the address of an element at A[3][3][3] using column-major ordering.

Clearly mention the expressions for the above ordering used in the computations. 5(CO1)

2. (A) Use stack to transform the following arithmetic expression into its equivalent postfix form :

A ^ (B + C) * (E + D / C) – F % B

Show the contents (stack frame) at each stage of conversion. 6(CO2)

(B) A circular queue, Q, has been created using a linear array to store numeric keys. Write algorithms for inserting a key into Q. Test your algorithms appropriately. 4(CO2)

3. (A) Consider a singly linked linear list. The node of the linked list is composed as – a numeric key value, DATA and a list node pointer, LINK. Write and test the algorithms to —

(1) Insert a node in the list with DATA as KEY.

(2) Reverse the linked list. 7(CO2)

(B) Discuss the specific advantages of doubly linked linear lists over singly linked linear lists. 3(CO2)

4. (A) State whether or not the binary tree in **Fig.-4A** is an AVL Tree. If it is an AVL tree, state reasons, otherwise transform the binary tree into an AVL tree.

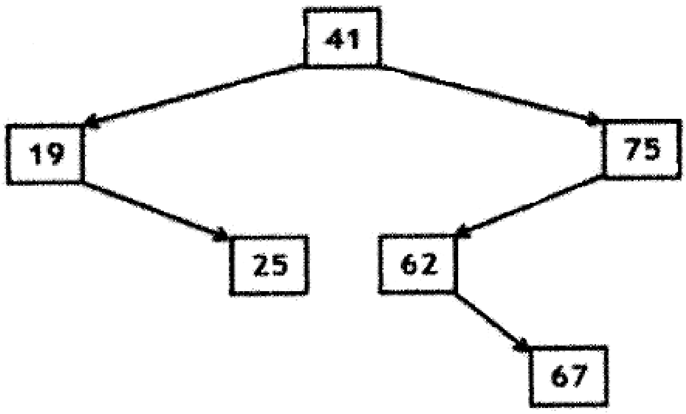

Fig.-4A

Now handle the following operations on the AVL tree – insert(20), insert(80), insert(90), insert(65) and delete(67).

Show step-by-step insertion / deletion stating AVL tree violations and associated actions. 6(CO4)

(B) Consider a rooted binary tree and write algorithms to —

(1) Count and print the parent nodes (exclude Root).

(2) Find height of the tree. 4(CO4)

5. (A) A set of keys are organized as a binary heap as shown in **Fig.-5A**. Order the key in ascending sequence using Heap sort. Show step-by-step process of ordering the keys.

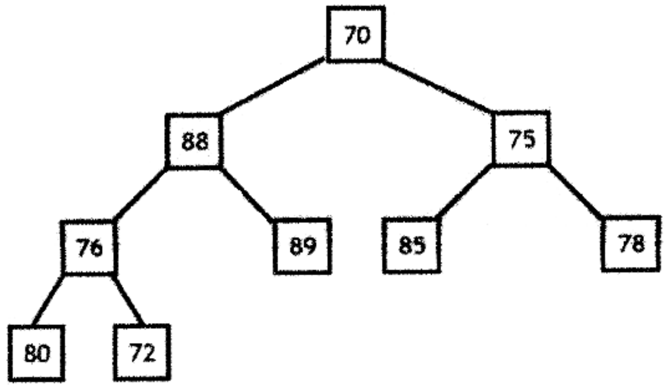

Fig.-5A 6(CO4)

(B) A hash table has 7 buckets, each with capacity to store 2 keys. Show step-by-step process of inserting following keys in the mentioned sequence into the hash table using –

(1) Linear probing, and

(2) Quadratic probing.

199, 67, 133, 89, 24, 54, 71, 45, 80, 25

Indicate specific collisions encountered and resolved in the process. 4(CO3)

6. (A) For the graph shown in **Fig.-6A**, sketch the DFS trees rooted at nodes C and F respectively. Also sketch the BFS tree rotated at C. The nodes will be processed in lexicographic order while traversing.

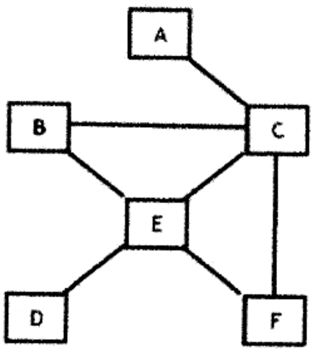

Fig.-6A,

Derive conclusions from the traversals for DFS and BFS. 3(CO4)

(B) For the graph **Fig.-6B**, apply Prim's algorithm for constructing (step-by-step) a minimum cost spanning tree. All edges are labeled with unit cost. The nodes will be processed in lexicographic sequence.

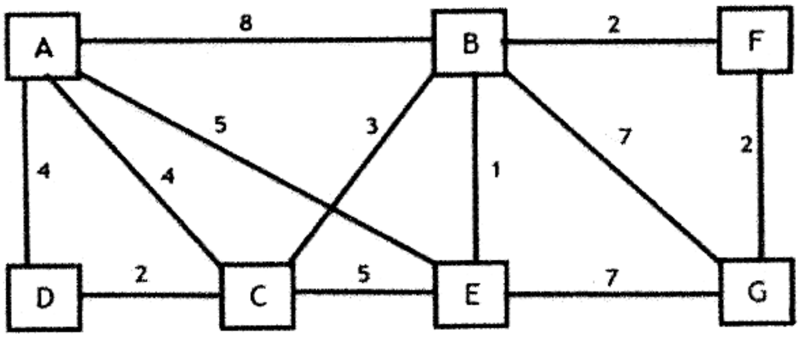

Fig.-6B 7(CO4)