Code No. 15049 / AICTE

FACULTY OF ENGINEERING

B.E. (I.T) III – Semester (AICTE) (Main & Backlog) Examination, July 2021

Time: 2 hours

Subject: Data Structures

Max. Marks: 70

Note: Missing data, if any may be suitably assumed.

PART – A

Answer any five questions.

(5x2 = 10 Marks)

1. Define Data Structure. Write the characteristics of Data Structures.
2. Write an ADT for an Array.
3. Define single, double and circular linked lists.
4. Write the postfix expression of $A * B * (C - D) / (E - F)$.
5. State the difference between complete binary tree and full binary tree.
6. When is an undirected graph said to be 'connected'?
7. Give the complexities for the following sorting algorithms
   a) Insertion sort   b) Merge sort   c) Quick sort   d) Heap sort
8. Define polymorphism and function overloading.
9. Write about Threaded Binary Tree with an example.
10. What is Hash Function? List few Hash Function.

PART – B

Answer any four questions.

(4x15 = 60 Marks)

11. Explain about space complexity and time complexity. Evaluate the time complexity for the following iterative function.

 | Line No. | Float sum (float *a const int n) |
    |----------|-----------------------------------|
    | 1 | } |
    | 2 | float s = 0 |
    | 3 | for (int i = 0 ; i < n ; i++) |
    | 4 | S+ = a[i] ; |
    | 5 | return s; |
    | 6 | } |

12. Write a C++ code to implement following operations on stack.
    a) Push   b) Pop   c) Display   d) Top

13. Define BST. Create a binary search tree with the following keys and perform in order, pre-order, post-order traversals on it.
    30, 20, 25, 40, 35, 36, 32, 45, 42

..2

14 Explain prim's algorithm and find minimum cost spanning tree for the following graph.
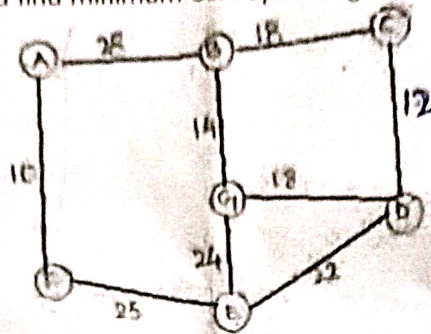
15 Explain the working of quick sort. Sort the following sequence of keys using quick sort 66, 77, 11, 88, 99, 22, 33, 44, 55. Show different passes (Trace) indicating the pivot and the partitions formed specify its time complexity.

16 Write an algorithm to add and subtract two polynomials using linked list.

17 Write short notes on following:
   a) Collision handling techniques in hashing
   b) AVL Trees

****