Code No: 15045/AICTE

FACULTY OF ENGINEERING

BE III – Semester (AICTE) (CSE) (Main & Backlog) Examination, July 2021

Subject: Data Structures & Algorithm

Time: 2 hours | Max. Marks: 70

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

PART – A

Answer any five questions. **(5x2 = 10 Marks)**

1. Define Space and Time complexity. Write the metrics needed for calculation.
2. How Dynamic memory allocation and deallocation are done in C++?
3. Define ADT. Write stack ADT.
4. What is the need for using circular arrays to implement queues?
5. Differentiate between single linked list and doubly linked list.
6. Evaluate the following postfix expression.
   6 2 3 + - 3 8 2 / + * 2 3 / +
   What is the stack top after evaluating the given expression?
7. Write the different binary tree traversals.
8. List out different representations of Graph.
9. What is hashing? What do you mean by collision in hashing?
10. Define max heap and min heap.

PART – B

Answer any four questions. **(4x15 = 60 Marks)**

11 a) Explain how virtual functions are used along with base class pointers to implement runtime polymorphism?
   b) What is Exception handling? How can we achieve it in C++?

12 a) Write a C++ program to implement stack ADT using linked list.
   b) What is the use of 'this' pointer?

13 a) Write a C++ program to insert and delete a node into/from a double linked list.
   b) Discuss the above operations time complexity.

14 a) Construct an AVL tree by inserting the following numbers in the order in which they are given. (Draw figure in each step)
   17 25 19 23 75
   b) How do you copy a binary tee? Explain the code with an example.

15 a) What is 'DIVIDE and CONQUER' strategy?
   b) Sort the following numbers using heap sort.
   5, 23, 7, 18, 2, 1, 9, 15, 6, 4, 8, 3, 13

16 a) What is BFS and DFS? Explain with an example.
   b) Explain Kruskal's algorithm to find minimum spanning tree.

17 a) Consider the hash function H(i)=(2i+5)%11.
   Insert keys 3, 8, 102, 23, 4, 10, 9, 12, 44, 15 and construct the 11 item hash table by using open addressing.
   b) Explain constructor and destructor with a program.

****