

DATA STRUCTURES
(Computer Science & Engineering)

Time: 3 hours

Max. Marks: 70

PART – A
(Compulsory Question)

- 1 Answer the following: (10 X 02 = 20 Marks)
- Define θ notation.
 - Write the procedure for deleting an element from the list.
 - Convert $((A + B) * C - (D - E)) \$(F + G)$ to postfix and prefix notation.
 - What are the limitations of linear queue? How they can be rectified?
 - What is complete binary tree?
 - What are the conditions for a graph to become a tree?
 - What is the time complexity of exchange sort?
 - Consider a situation where swap operation is very costly. Which sorting algorithm should be preferred so that the number of swap operations are minimized in general?
 - What is the recurrence relation for worst case of Binary Search?
 - Consider a hash table with 100 slots. Collisions are resolved using chaining. Assuming simple uniform hashing, what is the probability that the first 3 slots are unfilled after the first 3 insertions.

PART – B
(Answer all five units, 5 X 10 = 50 Marks)

UNIT – I

- 2 What are the asymptotic notations used? Explain.
- OR**
- 3 Write an algorithm to insert an element into a link list implemented by using arrays (assume already a linked list is in array is available which is available in alphabetical order).

UNIT – II

- 4 Show the detail content of the STACK for the evaluation of the following expression?
 $623 + -382 / + * 2\$3+$

OR

- 5 Implement a queue so that each element of a queue holds a list of integers. Write the functions add Q and remove Q from such queue

UNIT – III

- 6 Write a recursive program to perform in-order and post-order traversal of a binary tree.

OR

- 7 Write and explain Dijkstra algorithm for finding shortest path. Give an example.

UNIT – IV

- 8 Construct a procedure to sort N numbers using heap sort and construct a heap for the given values:
97, 53, 59, 25, 42, 50, 30, 15, 23, 32.

OR

- 9 Write an algorithm for merge sort. State the complexity to sort n numbers.

UNIT – V

- 10 Modify the binary search algorithm so that in case of unsuccessful search it returns the index i such that $k(i) < \text{key} < k(i+1)$.

OR

- 11 Write a program to implement extendible hashing. If the table is small enough to fit in main memory, how does its performance compare with open and closed hashing?