

Design and Analysis of Algorithms Lab

Programs:

1. Using a stack of characters, convert an infix string to a postfix string.
2. Implement polynomial addition using a single linked list
3. Implement insertion, deletion, searching of a BST, and display Pre order, Post order and In order traversal's
4. Implement linear search on integer array
5. Implement heap sort using a min heap.
6. Implement DFS routine in a connected graph
7. Implement BFS routine in a connected graph

8. Greedy

1. Implement Dijkstra's shortest path algorithm using BFS
2. Given a set of weights, form a Huffman tree from the weights and also find out the code corresponding to each weight.
3. Take a weighted graph as an input, find out one MST using Kruskal's algorithm
4. Take a weighted graph as an input, find out one MST using prim's algorithm
5. Given a set of weights, profits and an upper bound M – Find out a solution to the fractional Knapsack problem

9. Divide and Conquer

1. Implement binary search on an integer array
2. Write a quick sort routine for an array of integers
3. Implement merge sort for the array of integers
4. Implement Strassen's matrix multiplication algorithm for matrices whose order is a power of two.
5. Large integer multiplication

10. Back tracking

1. Pattern matching in a given text using Brute force approach.
2. N-Queen problem using Brute force approach.

11. Dynamic programming

- 1.** Fibonacci series using Dynamic programming
- 2.** Binary knapsack using Dynamic programming
- 3.** Longest Increasing Subsequence using Dynamic programming
- 4.** Chain matrix multiplication using Dynamic programming
- 5.** Min edit Distance using Dynamic programming
- 6.** Longest common subsequence using Dynamic programming
- 7.** Floyd-Warshall algorithm