

**SILIGURI INSTITUTE OF TECHNOLOGY**  
**LABORATORY ASSIGNMENTS**  
**Algorithm LAB (PCC-CS494)**  
**Computer Science & Engineering**  
**B.Tech. 2<sup>ND</sup> Year Second Semester**  
**Session: 2020-21**

**Day 1:**

- a) WAP to implement **LINEAR SEARCH** iterative as well as recursive method. Estimate the running time complexity for both the cases.
- b) WAP to implement **BINARY SEARCH** iterative as well as recursive method. Estimate the running time complexity for both the cases.

**Day 2:**

- a) WAP to implement **QUICK** sort using Divide and Conquer Strategy. Estimate the running time for different pivots.
- b) WAP to implement **MERGE** sort using Divide and Conquer Strategy. Estimate the running time.

**Day 3:**

- a) WAP to implement **HEAP** sort. Estimate the running time.
- b) WAP to implement **COUNTING** sort. Estimate the running time.

**Day 4:**

- a) WAP to solve **TOWERS-OF-HANOI** problem using recursion. Estimate the running time.
- b) Write two recursive programs to compute  $X^n$ , where both  $X$  &  $n$  are integers, one computes it in  $O(n)$  time & other in  $O(\log(n))$  time.
- c) WAP to find out  $n^{\text{th}}$  **FIBONACCI** number using recursion and calculate the running time complexity.

**Day 5:**

- a) WAP to **multiply a chain of matrices optimally** using iterative version of Dynamic Programming approach. Check your program for the sequence of 4 matrices  $\langle M1, M2, M3, M4 \rangle$  whose sequence of dimensions is  $\langle 10, 20, 50, 1, 100 \rangle$ .

**Day 6:**

- a) WAP to find the **Shortest path** between all the pairs of vertices of a given Graph, i.e. to implement **Floyd's algorithm**. Run your program on a graph whose length matrix representation is as follows:

0	7	5	$\infty$
$\infty$	0	7	6
$\infty$	$\infty$	0	$\infty$
4	1	11	0

**Day 7:**

- a) WAP to solve **4-Queens** Problem using Backtracking approach.  
 b) WAP to solve **Graph Coloring** problem using Backtracking approach.

**Day 8:**

- a) WAP to find the **MST** of a given graph using **Prim's algorithm**. Check your program on a Graph whose length matrix is given as follows.  
 b) WAP to find the **MST** of a given graph using **Kruskal's algorithm**. Check your program on a Graph whose length matrix is given as follows.

-	1	$\infty$	4	$\infty$	$\infty$	$\infty$
1	-	2	6	4	$\infty$	$\infty$
$\infty$	2	-	$\infty$	5	6	$\infty$
4	6	$\infty$	-	3	$\infty$	4
$\infty$	4	5	3	-	8	7
$\infty$	$\infty$	6	$\infty$	8	-	3
$\infty$	$\infty$	$\infty$	4	7	3	-

**Day 9:**

- a) WAP to implement **BFS** and **DFS** on a given graph where the graph is represented as an adjacency list form.

**Day 10:**

- a) WAP to implement **Knuth-Morris-Prat (KMP)** algorithm for pattern matching.