Course Code: ETCS-351

Algorithm Design and Analysis

List of Experiments

- **SEARCHING AND SORTING ALGORITHMS**

1. To implement Linear search and Binary search, compare and analyse its time complexities (Best case, Average case & Worst case).
2. To implement following algorithm using array as a data structure and analyse its time complexities (Best case, Average case & Worst case).
    a. Insertion sort
    b. Bubble sort
    c. Selection sort
    d. Quick sort
    e. Merge sort
    f. Heap sort and other sorting techniques

- **DIVIDE AND CONQUER TECHNIQUE**

3. To implement Strassen's algorithm for Matrix Multiplication and analyse its time complexity.

- **GREEDY ALGORITHMS**

4. To implement Activity Selection problem and analyse its time complexity.
5. To implement Task Scheduling problem and analyse its time complexity.
6. To implement Huffman Coding and analyse its time complexity.
7. To implement Fractional Knapsack problem and analyse its time complexity.

- **DYNAMIC PROGRAMMING ALGORITHMS**

8. To implement Matrix Chain Multiplication problem using iteration and analyse its time complexity.
9. To implement Matrix Chain Multiplication problem using recursion and analyse its time complexity.
10. To implement Memoized Matrix Chain Multiplication problem and analyse its time complexity and compare with the iterative and recursive methods.
11. To implement Longest Common Subsequence problem and analyse its time complexity.
12. To implement Optimal Binary Search Tree problem and analyse its time complexity.
13. To implement 0-1 Knapsack problem and analyse its time complexity.
14. To implement Binomial coefficient problem and analyse its time complexity.

- **GRAPH ALGORITHMS**

    - ***TRAVERSAL***

15. To implement Breadth First Search and analyse its time complexity.
16. To implement Depth First Search and analyse its time complexity.

    - ***GREEDY ALGORITHMS***

17. To implement Kruskal's algorithm for minimum cost spanning tree and analyse its time complexity.
18. To implement Prim's algorithm for minimum cost spanning tree and analyse its time complexity.
19. To implement Dijkstra's single source shortest path algorithm and analyse its time complexity in different cases.

❖ ***DYNAMIC PROGRAMMING***

20. To implement single source shortest path Bellman Ford algorithm and analyse its time complexity.
21. To implement all pair shortest path algorithm and analyse its time complexity.
22. To implement all pair shortest path Floyd Warshall algorithm and analyse its time complexity.

- **STRING MATCHING ALGORITHMS**

23. To implement naive string matching algorithm and analyse its time complexity.
24. To implement Rabin Karp algorithm and analyse its time complexity.
25. To implement Knuth Morris Pratt algorithm and analyse its time complexity.