Course Name: Design & Analysis of Algorithm Lab
Course Code: PCCCS494
Semester: 4
Contact Hours: L-T-P: 0-0-4
Credits: 2
Prerequisite: Data Structure & Algorithms Lab

Description of the Experiments

1. Given a sorted array and a number X, search two elements of the array such that their sum is X. Expected time complexity is O(n).

2. Implement Binary Search using Divide and Conquer.

3. Apply Binary Search on 2D NxM array (A) having numbers stored in non-decreasing order under row-major scanning.

4. A Bitonic Sequence is a sequence of numbers which is first strictly increasing then after a point strictly decreasing. A Bitonic Point is a point in the bitonic sequence before which elements are strictly increasing and after which elements are strictly decreasing. Find bitonic points in a bitonic sequence.

5. Apply Merge Sort to count inversion pairs in an array. Two elements a[i] and a[j] form an inversion pair if a[i] > a[j] and i < j. Example: The sequence 2, 4, 1, 3, 5 has three inversions (2, 1), (4, 1), (4, 3).

6. Implement a greedy algorithm to solve the fractional knapsack problem.

7. Find the second largest and second smallest number simultaneously in an array using Divide & Conquer Principle.

8. Given a sorted array and a number x, write a function that counts the occurrences of x in the array. Expected time complexity is O(logn).

9. Median of two sorted arrays: There are 2 sorted arrays A and B; each of size n. Write an algorithm to find the median of the array obtained after merging the above 2 arrays (i.e. array of length 2n). The complexity should be O(log(n)).

10. Given an array of digits, sort them with time complexity O(n).

11. Find neighbors of the median element in an array using the partitioning strategy of the Quick-Sorting method.

12. Given an array p[] which represents the chain of matrices such that the i-th matrix Ai is of dimension p[i-1] x p[i]. We need to write a function that should return the optimal parenthesizing expression resulting in a minimum multiplication cost to multiply the chain.

13. Given weights and values of n items, put these items in a knapsack of capacity W to get the maximum total value in the knapsack. You cannot break an item, either pick the item, or don't pick it.

14. Implement the greedy algorithm to solve the problem of the Job Sequencing with deadlines.

15. Implement a greedy algorithm for finding the single-source shortest paths. Suggest an algorithm if the given graph contains negative weights and non-negative weight cycle and implement it.

16. Apply Strassen's Matrix Multiplication strategy for odd dimensional square matrices.

17. Given a cost 2D-matrix and a position (m, n), write a function that returns the minimum cost-path to reach (m, n) from (0, 0).

18. Given a value V and an infinite supply of coins of m-denominations {C1=1<C2<C3<..<Cm}, we want to make change for Rs. V. Apply DP strategy to find out the minimum number of coins to make the change?

19. Given a set of non-negative integers, and a value sum, determine if there is a subset of the given set with sum equal to given sum.

20. Implement DP strategy to solve the Traveling Salesman Problem (TSP).

21. Implement all pairs of the Shortest path algorithms for a graph using Floyd Warshall's strategy.

22. Students need to develop a software or tool using any language for Plagiarism Checker. The primary objective of this project is that they have to implement a data structure concept and algorithm and show us how they implement it.

23. Professor Sarkar thinks he has discovered a remarkable property of binary search trees. Suppose that the search for key k in a binary search tree ends up in a leaf. Consider three sets: A, the keys to the left of the search path; B, the keys on the search path; and C, the keys to the right of the search path. Professor Bunyan claims that any three keys $a \in A$, $b \in B$, and $c \in C$ must satisfy $a \leq b \leq c$. Give a smallest possible counterexample to the professor's claim.

24. KMP String Matching: Given a text txt[0..n-1] and a pattern pat[0..m-1], write a function search(char pat[], char txt[]) that prints all occurrences of pat[] in txt[]. You may assume that n > m.

Text: A A B A A C A A D A A B A A B A

Pattern: A A B A

25. Implement a routine management system that will work for the next semester.