10B17CI471: Algorithms Lab

Course Credit: 1

Semester: IV

Objective:

1. To learn the notions of Algorithm, Models of computation.
2. To learn Integer arithmetic, Square roots, Newton's method, Euclids method
3. To learn Evaluation of algorithm: Computational complexity, order notations, recurrences, and Master's theorem.
4. To learn Sorting: Insertion sort, merge sort, Heaps and heap sort, Quick sort, Linear sort, priority queue, order statistics, lower bounds for sorting
5. To learn Searching: Balanced tree, red-black tree, Skip list, Hashing, universal and perfect hashing, lower bounds for searching
6. To learn Graph: representation and algorithms, Breadth-first search (BFS), Depth-first search (DFS), topological sorting, Shortest Paths, Single-source shortest paths problem, Dijkstra, Bellman-Ford
7. To learn Dynamic Programming techniques: sub problems, guessing, bottom-up; Fibonacci, shortest paths, Parent pointers; text justification, perfect-information blackjack , String sub problems, psuedopolynomial time; parenthesization, edit distance, knapsack
8. To learn P –NP: classes of P and NP, NP-hard, NP-complete, reduction
9. To learn Optimality and approximation algorithms

Learning Outcomes:

1. Application to Job scheduling in OS
2. Application to Routing in network
3. Application to Range assignment, TSP
4. Application to Online games Chess, Sudoku, CCS etc.

List of Experiments

S NO	Topics
1	Write c programs for 1) Prime number (check the given number is prime or not) 2) GCD of k numbers 3) Fibonacci sequence [f(n) = f(n-1) + f(n-2); n > 2, f(1)=1, f(2)=1] • Compute running time using time library functions. • Prepare table for input size vs. running time. For same input size note best run and worst run of your experiments.
2	Write c programs for 1) Insertion Sort of n random number. 2) Merge Sort of n random number 3) Heap Sort of n random number Where n is sufficiently large. 4) Implement Priority Queue using Heap. • Compute running time. • Prepare table for number of elements vs. running time.
3	A) Implement the following variations of Quicksort using an array:

	1. First element is the partition element 2. Last element is the partition element 3. Randomized QuickSort (Take the partition element at random) B) Write a program to find k^{th} order statistics.
4	Write c programs for 1) Constructing a red-black tree of n random number and compute the average time of insertion, deletion and searching. 2) Constructing a skip-list of n random numbers and compute the average time of insertion, deletion and searching. Where n is sufficiently large. Prepare table for number of elements vs. average time of insertion, deletion and searching.
5	
6	1) Implement Prim's algorithm to find MST of a weighted graph. 2) Implement disjoint set data structures. 3) Implement Kruskal algorithm using above data structure..
7	1) Implement Dijkstra algorithm to find shortest path of a weighted graph. 2) Implement a FordFulkerson algorithm to find max flow of a flow network.
8	1) Implement Rabin Krap algorithm for pattern matching 2) Implement a Patricia for text searching
9	Write c programs for the following problems using Dynamic programming methods. 1) Longest common subsequence 2) Matrix chain multiplication 3) Coin denomination 4) Shortest path 5) Knapsack 6) E.t.c……
10	1) Implement back tracking algorithm to find a solution of the 3-SAT problem. 2) Implement branch and bound algorithm to find a solution of TSP.

References

1. Aho, Hopcraft, Ullman : Data Structures and Algorithms
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduction to Algorithms,
3. S. Dasgupta, C. Papadimitrou, U. Vazirani: Algorithms

Evaluation Scheme:

1.	Mid Term Exam (Viva and Written Exam)	20
2.	End term Exam (Viva and Written Exam)	30
3.	Lab Records	5
4.	Regular Assessment (Quality and quantity of experiment performed, Learning laboratory skills, Attendance etc.)	30
5.	Project	15
	Total	**100**