Department of Computer Science and Engineering
International Institute of Information Technology, Naya Raipur
Design and Analysis of Algorithms (Course Code: CSE201)

Time: 24:00 hrs (12 Weeks) Lab Experiments Maximum Marks: 300

Note: Implement the following Algorithms in C/C++/Java/Python and submit the same by Lab Record

1. Divide−and−Conquer
   - 1.1 Binary Search (Recursive and Iterative)
   - 1.2 Merge Sort
   - 1.3 Quick Sort
   - 1.4 Given a sorted array of non-repeated integers $A[1...n], n > 1$ then check whether there is an index i for which $A[i] = i$. Give an algorithm that runs in $O(logn)$ time [40]
2. Divide−and−Conquer
   - 2.1 Strassen′s Matrix Multiplication [20]
3. Miscellaneous Examples based on Divide and Conquer Algorithms
   - 3.1 Given an array of n elements. Find whether there are two elements in the array such that their sum is equal to given element K or not? in $O(nlogn)$ time.
   - 3.2 Given an array of n elements. Find whether there are three elements in the array such that their sum is equal to given element K or not? in $O(n^2)$ time.
   - 3.3 Let A and B be two arrays of n elements. Given a number K, draw an $O(nlogn)$ time algorithm for determining whether there exists $a \in A$, $b \in B$ such that $a + b = K$ or not?.
   - 3.4 Given an array of n elements, give an algorithm for checking whether there are any duplicate elements in the array or not? in $O(nlogn)$ time.
   - 3.5 Given an array of n elements, give an algorithm for finding the element which appears maximum number of times in the array in $O(nlogn)$ time. [50]
4. Greedy Method
   - 4.1 Knapsack Problem
   - 4.2 Job sequencing with deadlines algorithm
   - 4.3 Prim′s Algorithm for finding the minimal spanning trees
   - 4.4 Krushkal′s Algorithm for finding the minimal spanning trees
   - 4.5 Dijkstra′s Algorithm [50]
5. Dynamic Programming
   - 5.1 Finding the optimal order of multiplying n matrices
   - 5.2 Construction of OBST
   - 5.3 0/1 Knapsack Problem
   - 5.4 All pairs shortest path problem

5.5 Traveling Salesmen Problem [60]

6. Back Tracking

6.1 Demonstrate N-Queens Problem

6.2 Detecting Hamiltonian Cycle

6.3 Demonstrate Graph Coloring Problem

6.4 Demonstrate Sum of subsets Problem [80]

Note: Every one has to submit the record (Indentation in the program is must) with detailed solution and it's worst case time complexity.

Expected Internal Evaluation Milestones

Week-III:	1.1, 1.4, 3.1
Week-IV:	3.2, 3.3, 3.4
Week-V:	3.5, 1.2, 1.3
Week-VI:	2.1
Week-VII:	4.1, 4.2
Week-VIII and Week-IX:	4.3, 4.4, 4.5
Week-X:	5.1, 5.2
Week-XI and Week XII:	5.3, 5.4, 5.5
Week-XIII and Week XIV:	6.1, 6.2, 6.3, 6.4