RAISONI GROUP
a vision beyond

G. H. Raisoni College of Engineering
and Management, Wagholi, Pune.

Department of
Artificial Intelligence

Lab Manual (2022-23)

Course: Design and Analysis of Algorithm (DAA)

Class: TY AI Term: V

Faculty: Prof. Bhagyashri Wankar

G. H. Raisoni College of Engineering and Management, Wagholi, Pune.

Department of Artificial Intelligence

Course Details

Course: DAA Lab

COURSE OUTCOME	
CO1	Recall basic concepts of algorithm in analysis and Design of algorithms.
CO2	Examine Recurrence relations, solutions of recurrence of searching sorting methods.
CO3	Analyze Greedy methods used for analysis and Design of Algorithm.
CO4	Apply Dynamic Programming concepts in designing algorithm.
CO5	Evaluate advanced techniques and tools available for algorithm analysis and development.

List of Experiments

Sr. No.	Experiment List	CO Mapping	Software Required
1	Write C++ program to find factorial of a given number using (i) Recursion (ii) Iteration Compare time and space complexity of both the designs.		Any C++ Compiler
2	Implement Binary search program with Divide and Conquer design strategy for n numbers using C++. Discuss Best, Average and Worst time complexity.		Any C++ Compiler
3	Sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of n and record the time taken to sort. The elements can be read from a user or can be generated using the random number generator.		Any C++ Compiler

	Demonstrate using C++ how the divide and conquer method works along with its time complexity analysis: worst case, average case and best case.		
4	A business house has several offices in different countries; they want to lease phone lines to connect them with each other and the phone company charges different rent to connect different pairs of cities. Business house want to connect all its offices with a minimum total cost. Solve the problem by suggesting appropriate data structures in C++.		Any C++ Compiler
5	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm. Write the program in C++.		Any C++ Compiler
6	Implement a program in C++ for 0/1 Knapsack problem using Dynamic Programming method.		Any C++ Compiler
7	Write C++ program to implement Travelling Sales Person problem using Dynamic programming.		Any C++ Compiler
8	Implement C++ program for solving N-Queen‟s problem using Back tracking. (Assume N=4)		Any C++ Compiler
9	Implement Travelling salesman problem using branch and bound approach using C++.		Any C++ Compiler
10	Write C++ Program to demonstrate the implementation of Rabin-Karp Algorithm with discussion of time complexity.		Any C++ Compiler
	Content Beyond Syllabus		
1	Matrix Chain Multiplication using Dynamic Programming		Any C++ Compiler
2	Case Study of Optimization Algorithms for complexity problems		Any C++ Compiler

Assignment No. 1

Aim: Write C++ program to find factorial of a given number using (i) Recursion (ii) Iteration Compare time and space complexity of both the designs.

Objectives:

1. To take a positive integer from user and calculate the factorial of that number.
2. To calculate time and space complexity of an algorithm.

Theory:

Factorial of a non-negative integer n is the product of all the positive integers that are less than or equal to n.

For example: The factorial of 6 is 720.

6! = 6 * 5 * 4 * 3 * 2 *1

6! = 720

The factorial of an integer can be found using a recursive program or an iterative program.

A for loop can be used to find the factorial of a number using an iterative program.

Algorithm 1: To calculate factorial of a number using recursive function.

```
Step 1: Start
Step 2: Read number n
Step 3: Call factorial(n)
Step 4: print factorial f
Step 5: stop
        factorial (n)
        Step 1: if n = 1 return 1
        Step 2: else
                f = n * factorial (n - 1)
        Step 3: return f
```

Algorithm 2: To calculate factorial of a number using iterative method.

```
Step 1: Start
Step 2: Read number n
Step 3: Set i=1, fact=1
Step 4: while ( i <= num)
        fact = fact * i
        i = i + 1
Step 5: Print factorial f
Step 6: Stop
```

Program 1:

Output:

Program 2:

Output:

Conclusion: