**SVKM’s NMIMS**

**School of Technology Management & Engineering**

**Computer Engineering Department**

Program: B.Tech/MBA Tech Sem II

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**Experiment No.06**

**A.1 Aim:**

To understand the concept of computational complexity and Recursion

|  |  |  |
| --- | --- | --- |
| **Algorithm steps** | No. of inputs | Time |

**Task 1: Complete the table and compute the running time for the following functions**

1. Finding the sum of elements of an array
2. Addition of two matrices

**Task 2**: Define and explain asymptotic notations

**Task 3:** Implement the following functions using recursion

a) Factorial of a number b) Fibonacci series

**A.2 Prerequisite:**

1. Knowledge of complexity analysis

2. Knowledge of Array Handling.

**A.3 Outcome:**

**After successful completion of this experiment students will be able to**

1. Understand the concept of computational complexity
2. Understand and implement the concept of recursion

**A.4 Theory:**

In theoretical analysis of algorithms, it is common to estimate their complexity in the asymptotic sense, i.e., to estimate the complexity function for arbitrarily large input. The term "analysis of algorithms" was coined by Donald Knuth. Algorithm analysis is an important part of computational complexity theory, which provides theoretical estimation for the required resources of an algorithm to solve a specific computational problem. Most algorithms are designed to work with inputs of arbitrary length. Analysis of algorithms is the determination of the amount of time and space resources required to execute it. Usually, the efficiency or running time of an algorithm is stated as a function relating the input length to the number of steps, known as **time complexity**, or volume of memory, known as **space complexity**

**Complexity of Algorithms**

The complexity of an algorithm M is the function f(n) which gives the running time and/or storage space requirement of the algorithm in terms of the size ‘n’ of the input data. Mostly, the storage space required by an algorithm is simply a multiple of the data size ‘n’.

Analysis of algorithm is the process of analyzing the problem-solving capability of the algorithm in terms of the time and size required (the size of memory for storage while implementation). However, the main concern of analysis of algorithms is the required time or performance.

Complexity shall refer to the running time of the algorithm. The function f(n), gives the running time of an algorithm, depends not only on the size ‘n’ of the input data but also on the particular data.

The complexity function f(n) for certain cases are:

1. Best Case: The minimum possible value of f(n) is called the best case.

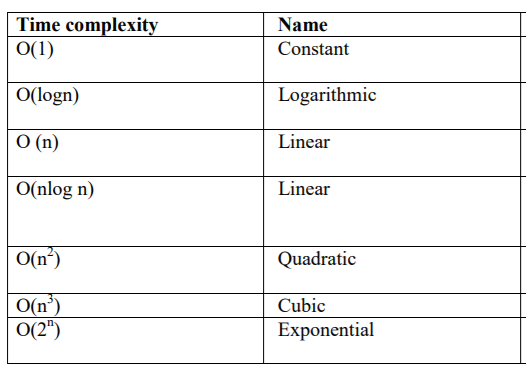
2. Average Case: The expected value of f(n).

3. Worst Case: The maximum value of f(n) for any key possible input.

**Asymptotic Analysis**

The asymptotic behavior of a function (𝒏) refers to the growth of 𝒇(𝒏) as n gets large. We typically ignore small values of n, since we are usually interested in estimating how slow the program will be on large inputs.

Asymptotic complexity measure means that we don’t try to count the exact number of steps of a program, but how that number grows with the size of the input to the program. That gives us a measure that will work for different operating systems, compilers and CPUs. The asymptotic complexity is written using big-O notation. · It is a way to describe the characteristics of a function in the limit. · It describes the rate of growth of functions. · Focus on what’s important by abstracting away low-order terms and constant factors. · It is a way to compare “sizes” of functions:

****

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments within two hours of the practical.***

***The soft copy must be uploaded on the Portal.)***

|  |  |
| --- | --- |
| Program: | Sem: |
| Roll No. | Name: |
| Division: | Batch : |
| Date of Experiment: | Date of Submission: |
| Grade : |  |

**B.1 Software Code written by student:**

***(Paste your code completed during the 2 hours of practical in the lab here)***

**B.2 Input and Output:**

***(Paste your commented program input and output in following format, If there is error then paste***

***the specific error in the output part. In case of error with due permission of the faculty extension***

***can be given to submit the error free code with output in due course of time.)***

**B.3 Observations and learning:**

***(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)***

**B.4 Conclusion:**

*(****Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)***

**B.5 Question of Curiosity**

***(To be answered by student based on the practical performed and learning/observations)***

Q. Is theoretical approach better than experimental approach to calculate computational complexity. Justify

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*