 

**Faculty of Technology and Engineering Chandubhai S Patel Institute of Technology Department of Computer Science & Engineering**

Date: / / 2024

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Academic Year | : | 2024-25 | Semester | : | 4 |
| Course code | : | CS358 | Course name | : | Design and Analysis of Algorithms |

**Performa for PRACTICAL – 1 Program Analysis Guidelines:**

1. Implement a problem in any language.
2. Count the no of instructions of the program for at least different **10 randomly generated input**
3. Make a table of **Input Size Vs No of Instructions**
4. Draw the Line/ Scatter Chart of **Input size Vs No of Instructions**
5. Repeat step 2 to 4 for **Best Case, Average Case, and Worst Case if Applicable**
6. Compare the **practical complexity with the theoretical complexity**.

~~7. Write the Conclusion from Graph and/or Data Table~~

**Aim**: Implement and analyze algorithms using iterative and recursive approaches for the problems given below

1. **Algorithm & Time Complexity :**
2. **Program with Counter of Primitive Operations and Comments :**
3. **Scatter Chart with Equation**:
   1. Input Size vs. Primitive Operation

**Data Table1 Factorial iterative :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Size**  **(To be generated using random function only)** | **No. of Primitive Operation** | **Theoretical Complexity** | **Practical Complexity with Equation and Growth Rate** |
| 10 | 36 | O(n) | 3n + 6 |
| 15 | 51 |
| 20 | 66 |
| 50 | 156 |
| 100 | 306 |
| 1000000 | 3000006 |

**Data Table 2 Factorial recursive :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Size**  **(To be generated using random function only)** | **No. of Primitive Operation** | **Theoretical Complexity** | **Practical Complexity with Equation and Growth Rate** |
| 100 | 403 | O(n) | 4n + 3 |
| 150 | 603 |
| 200 | 803 |
| 500 | 2003 |
| 1000 | 4003 |
| 1000000 | 4000003 |

**Data Table 3 Fibonacci iterative :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Size**  **(To be generated using random function only)** | **No. of Primitive Operation** | **Theoretical Complexity** | **Practical Complexity with Equation and Growth Rate** |
| 100 | 700 | O(n) | 7n |
| 150 | 1050 |
| 200 | 1400 |
| 500 | 3500 |
| 1000 | 7000 |
| 1000000 | 7000001 |

**Data Table 4 Fibonacci recursive :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Size**  **(To be generated using random function only)** | **No. of Primitive Operation** | **Theoretical Complexity** | **Practical Complexity with Equation and Growth Rate** |
| 100 | 3.16 X (10^21) | O(2^n) | 4 X (2^n) |
| 150 | 7.8 X (10^31) |
| 200 | 2.05 X (10^42) |
| 500 | 1.24 X (10^105) |
| 1000 | 4.04 X (10^209) |
| 1000000 | Order of (10^208988) |

**Conclusion:**

**#Once the practical is completed in the lab, students will write answers to the following questions during the first 15 minutes of the next lab. (handwritten)**

**Related Questions:**

1. What are the limitations of using recursion for large inputs in the Fibonacci sequence and factorial problems?
2. Modify the recursive factorial function to use tail recursion. Analyze its efficiency compared to the standard recursive function.
3. Why is the recursive Fibonacci not preferable over iterative Fibonacci? Provide some real-time application of the Fibonacci sequence.
4. Compare the time and space complexities of iterative and recursive approaches for both problems. Use a table to summarize your findings.

**Grade / Marks Sign of Lab Teacher with Date**