Department of Computer Science and Engineering
Chandpur Science and Technology University

Course Code: CSE 2201	Credits: 1.50
Course Name: Algorithm Design and Analysis	Semester: 2-2

Lab 01

Introduction to Algorithm Design & Complexity Analysis

I. Learning Objectives

By the end of this lab, students should be able to:

- Define what constitutes an algorithm.
- Identify input size and step count.
- Differentiate between constant, linear, logarithmic, quadratic complexities.
- Apply Big-O, Big-Ω, Big-Θ to real code.
- Perform basic complexity analysis of algorithms manually.

II. Lesson Fit:

Prerequisite: C/C++, Data Structure

III. Theory Recap:

What is an Algorithm?

An **algorithm** is a finite sequence of well-defined steps for solving a problem.

Properties of an Algorithm:

Property	Description
Input	Receives zero or more inputs
Output	Produces at least one output
Definiteness	Clear, unambiguous instructions
Finiteness	Terminates after a finite number of steps
Effectiveness	Basic enough to perform by hand/machine

Asymptotic Notations

Notation	Represents	Definition
$O(f(n))$	Upper bound	Worst-case time: $T(n) \leq c \cdot f(n)$ for large n
$\Omega(f(n))$	Lower bound	Best-case time: $T(n) \geq c \cdot f(n)$
$\Theta(f(n))$	Tight bound	Average-case: $T(n) = c \cdot f(n)$

Lab 1 Activity List

Experiment # 1: *Linear Search & Step Analysis*

📊 Time Complexity:

- Best Case: O(1) → Key at index 0
- Worst Case: O(n) → Key not present
- Average Case: O(n)

🧮 Space Complexity: O(1) (no extra memory used)

Experiment # 2: Binary Search (Requires Sorted Array)

📊 Time Complexity:

- Best Case: O(1)
- Worst Case: O(log n)

🧮 Space Complexity: Space Complexity: O(1)

📊 Comparison Table (Empirical)

n (array size)	Linear Steps (Worst)	Binary Steps (Worst)
10	10	~4
100	100	~7
1,000	1000	~10
10,000	10000	~14
100,000	100000	~17

You can plot this in a graph (X: Input size n, Y: Steps taken) to **visually compare** the growth rate.

Experiment # 3: Bubble Sort – Complexity Analysis

📊 Time Complexity:

- **Best: O(n) — if already sorted with optimization**
- **Worst: $O(n^2)$**
- **Average: $O(n^2)$**

🧮 Space Complexity: O(1)

Report:

The report should cover the following

Name of the Experiment

1. Objective
2. Algorithm
3. Theoretical Solution of given problem
4. Practical Work:
    a. Pseudocode
    b. Source Code in C/CPP/Python/Matlab
5. Analysis Table

Algorithm	Best Case	Worst Case	Avg Case	Space
Linear Search	O(1)	O(n)	O(n)	O(1)
Binary Search	O(1)	O(log n)	O(log n)	O(1)
Bubble Sort	O(n)	$O(n^2)$	$O(n^2)$	O(1)

6. Observations
    - Binary search is much faster but only works on sorted data.
    - Linear search is more flexible but inefficient for large arrays.
    - Step counting connects theoretical Big-O with practical execution.
7. Challenges
    - Remembering loop bounds during nested iteration analysis.
    - Debugging while counting execution steps accurately.
8. Conclusion

📷 Attachments:

- Screenshot of program output.

- Manual step count snapshots.

- Complexity graph (drawn or plotted).