Notion of an algorithm, Fundamentals of algorithmic problem solving and problem types

Session No.: 1
Course Name: Design and analysis of algorithm
Course Code: R1UC407B
Instructor Name: Dr. Mili Dhar
Duration: 50 Min.

Review of Data, Data Structure and Algorithm

In computer science, **data**, **data structures**, and **algorithms** form the foundation of efficient problem-solving.

1. Data represents raw facts and figures
2. Data structures provide a way to organize and store this data.
3. Algorithms define step-by-step procedures to process and manipulate it effectively.

Understanding the interplay between these concepts is crucial for optimizing computational tasks, improving performance, and developing robust applications.

Review of Data, Data Structure and Algorithm

Data

Data is the fundamental unit of information that computers process. It can be categorized into:

- **Primitive Data Types**: Integers, floating points, characters, and booleans.
- **Non-Primitive Data Types**: Arrays, structures, files, and objects.

Data Structures

A **data structure** is a specialized format for organizing, managing, and storing data efficiently. It determines how data is accessed, manipulated, and stored in memory. Data structures are broadly classified into:

1. **Linear Data Structures**
2. **Non-Linear Data Structures**

Review of Data, Data Structure and Algorithm

Algorithms

An **algorithm** is a finite set of well-defined instructions to solve a problem. The efficiency of an algorithm is measured by its **time complexity** (execution time) and **space complexity** (memory usage).

Open Question:

Have you ever followed a recipe to cook or a set of instructions to assemble a toy?

Learning Outcomes

LO1:Define an algorithm as a step-by-step procedure for solving problems.

LO2: Apply algorithmic thinking to real-world problems.

Session Outline

1 Introduction to notion of an algorithm,

2 Fundamentals of algorithmic problem solving and problem types

3 Concept and definition for learning Activity

4 Activity (scenario based)

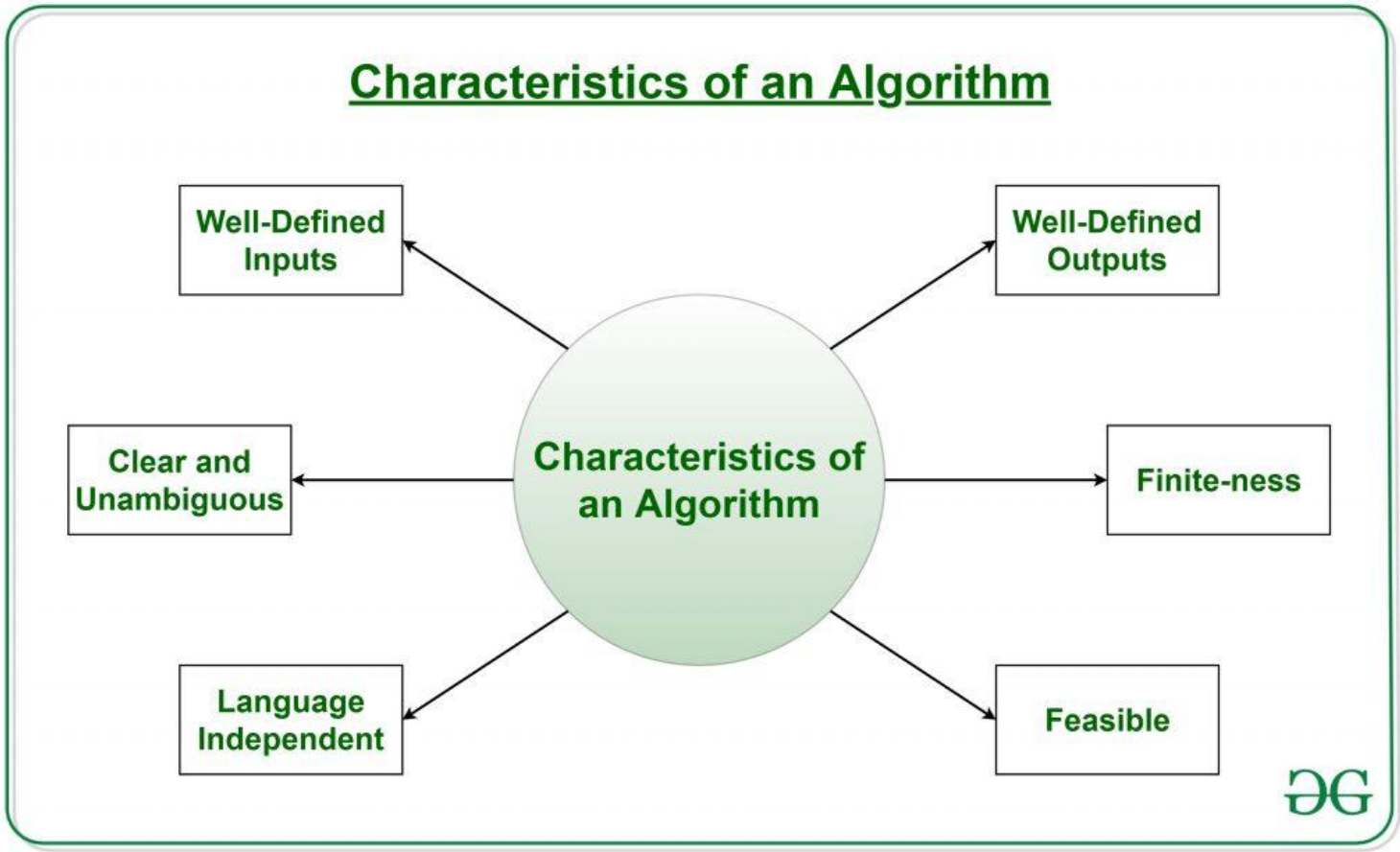
5 Reflection learning activity

6 Conclusion

Definition of Algorithm

- An **algorithm** is a step-by-step procedure or a well-defined set of rules used to solve a specific problem or perform a computation.

Characteristics of an algorithm



Characteristics of an algorithm

An effective algorithm must have the following properties:

1. **Well-defined Inputs and Outputs** – It takes some input, processes it, and produces a well-defined output.
2. **Finiteness** – It must complete in a finite number of steps.
3. **Definiteness** – Each step must be clear and unambiguous.
4. **Effectiveness** – Every step should be feasible and executable within a reasonable time.
5. **Generality** – It should solve a class of problems rather than a single instance.

Example 5, 8, 2, 32, 12, 9, 11, 3

Algorithm of linear search :

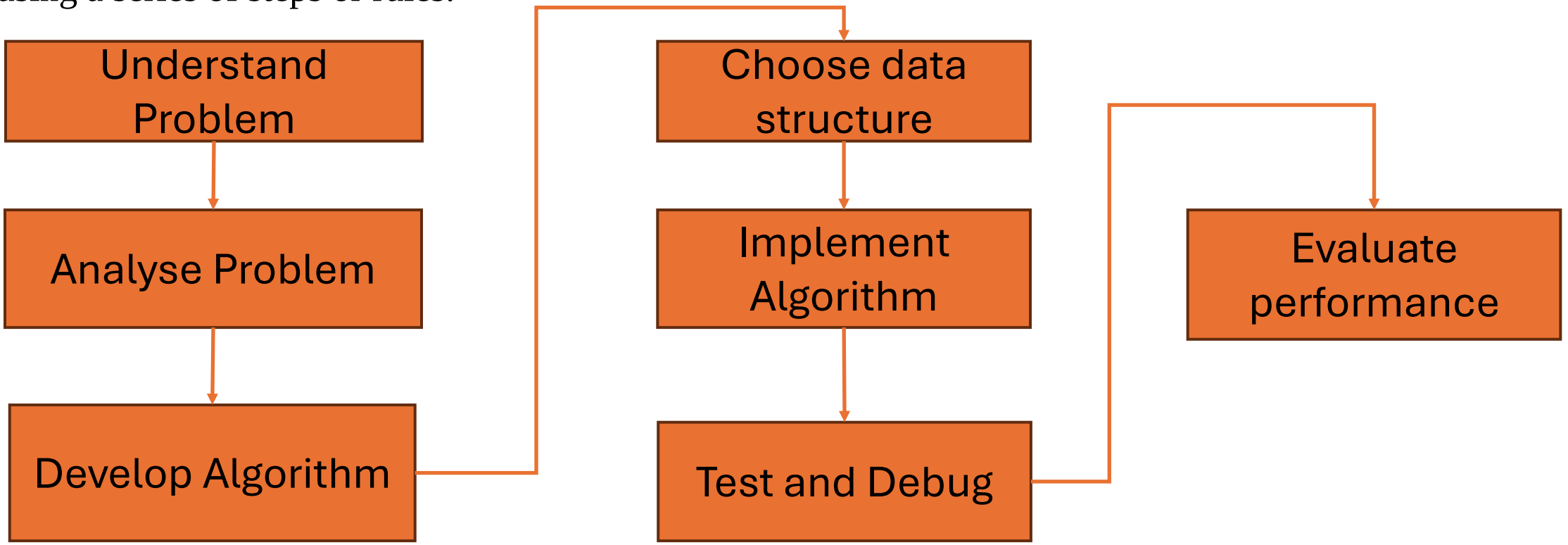
- Start from the leftmost element of arr[] and one by one compare x with each element of arr[].
- If x matches with an element, return the index.
- If x doesn't match with any of elements, return -1.

Program for Linear Search :

```
int search(int arr[], int n, int x)
{
    int i;
    for (i = 0; i < n; i++)
        if (arr[i] == x)
        return i;
    return -1;
}
```

Fundamentals of Algorithmic Problem Solving

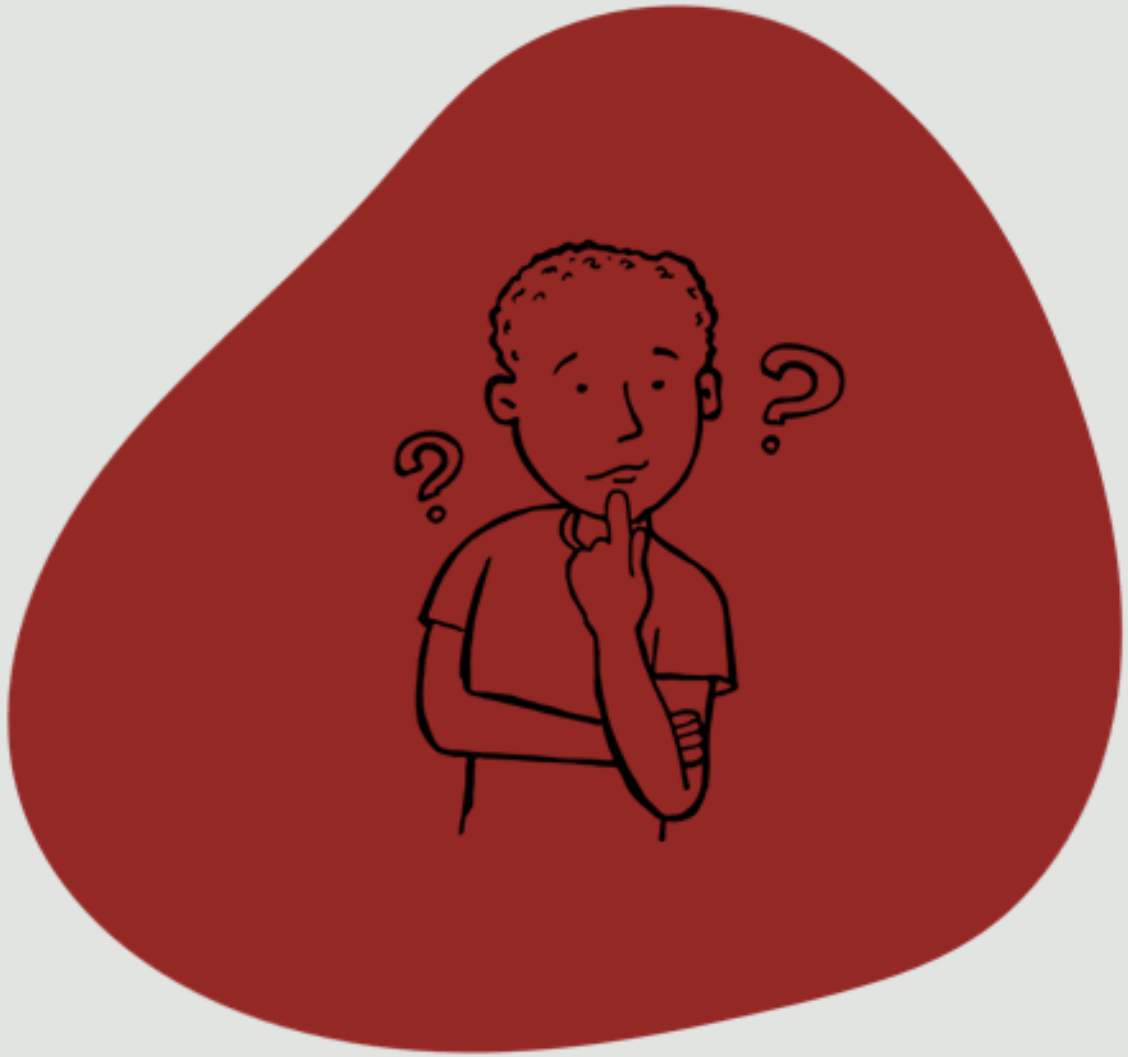
- Algorithmic problem-solving is the process of designing and implementing a solution to a problem using a series of steps or rules.



Algorithm	Programming
Written in design phase	Written in Implemented phase
Need domain knowledge	Need programmer knowledge
Can be written in any language	Can be written in programming language
Independent of hardware and software	Dependent on hardware and software
We analyse algorithm for time and space	We tests programs for faults

Activity 1:

Write the difference between Algorithm and Programming?



Pseudocode for Linear Search :

```
FUNCTION linearSearch(list, searchTerm):
    FOR index FROM 0 -> length(list):
        IF list[index] == searchTerm THEN
            RETURN index
        ENDIF
        ENDLOOP
            RETURN -1
END FUNCTION
```

Activity 2

Create a flowchart for linear search.

Summary

1. An algorithm is a sequence of computational steps that transform the input into output.
2. Once we find an algorithm for solving a problem, we do not need to re-discover it the next time we are faced with that problem.
3. All the knowledge required for solving the problem is present in the algorithm. you don’t have to spend time for rethinking it in future.
4. Makes it easy when explaining the process to others
5. Step by step instructions or flowchart or pseudocode are the ways to represent an algorithm for a program.
6. pseudocode does not have a specific syntax like any of the programming languages and thus cannot be executed on a computer.
7. Difference between algorithm and programming

In the next session, analysis of algorithm efficiency, asymptotic notation will be discussed in detail.