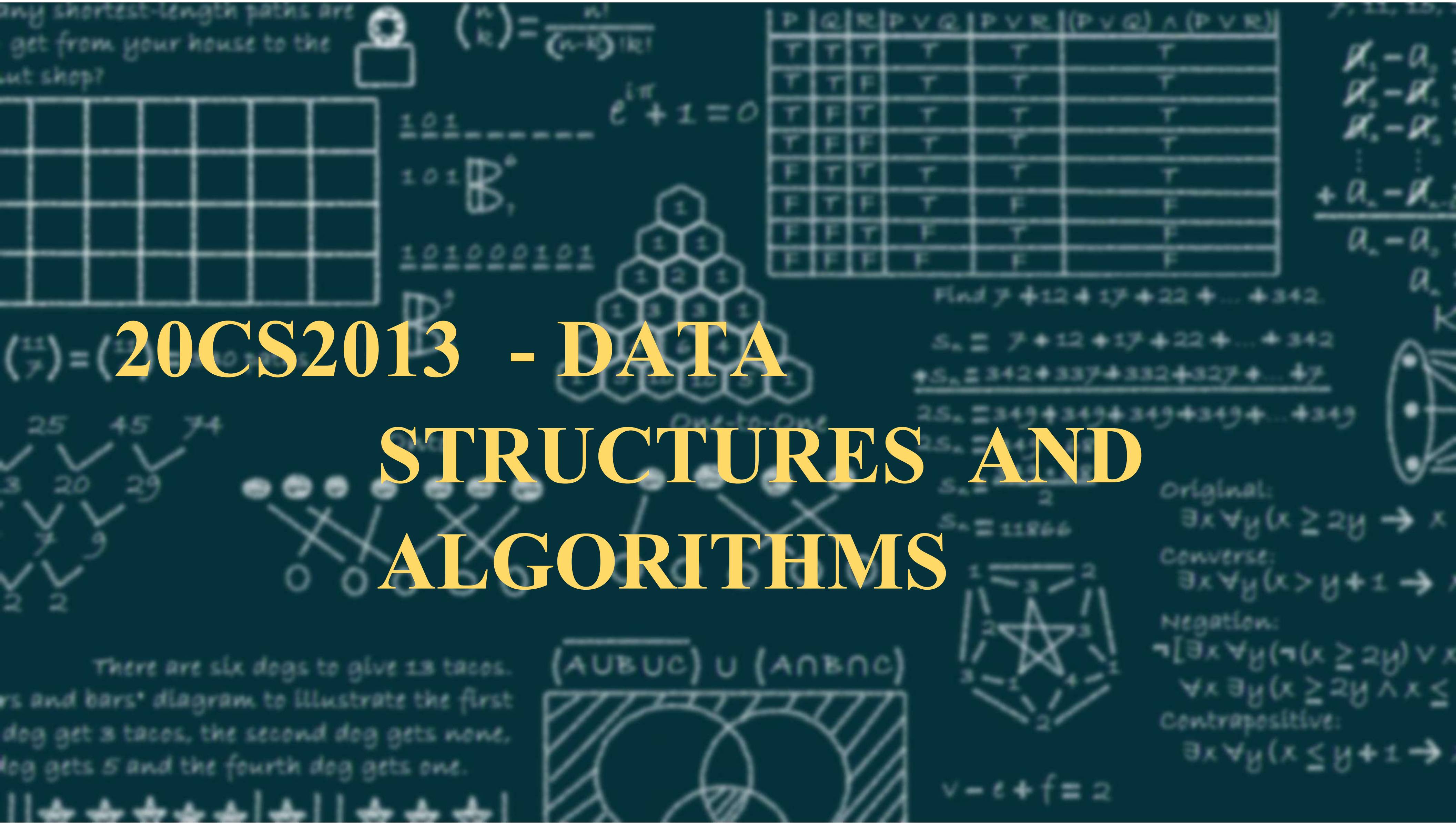
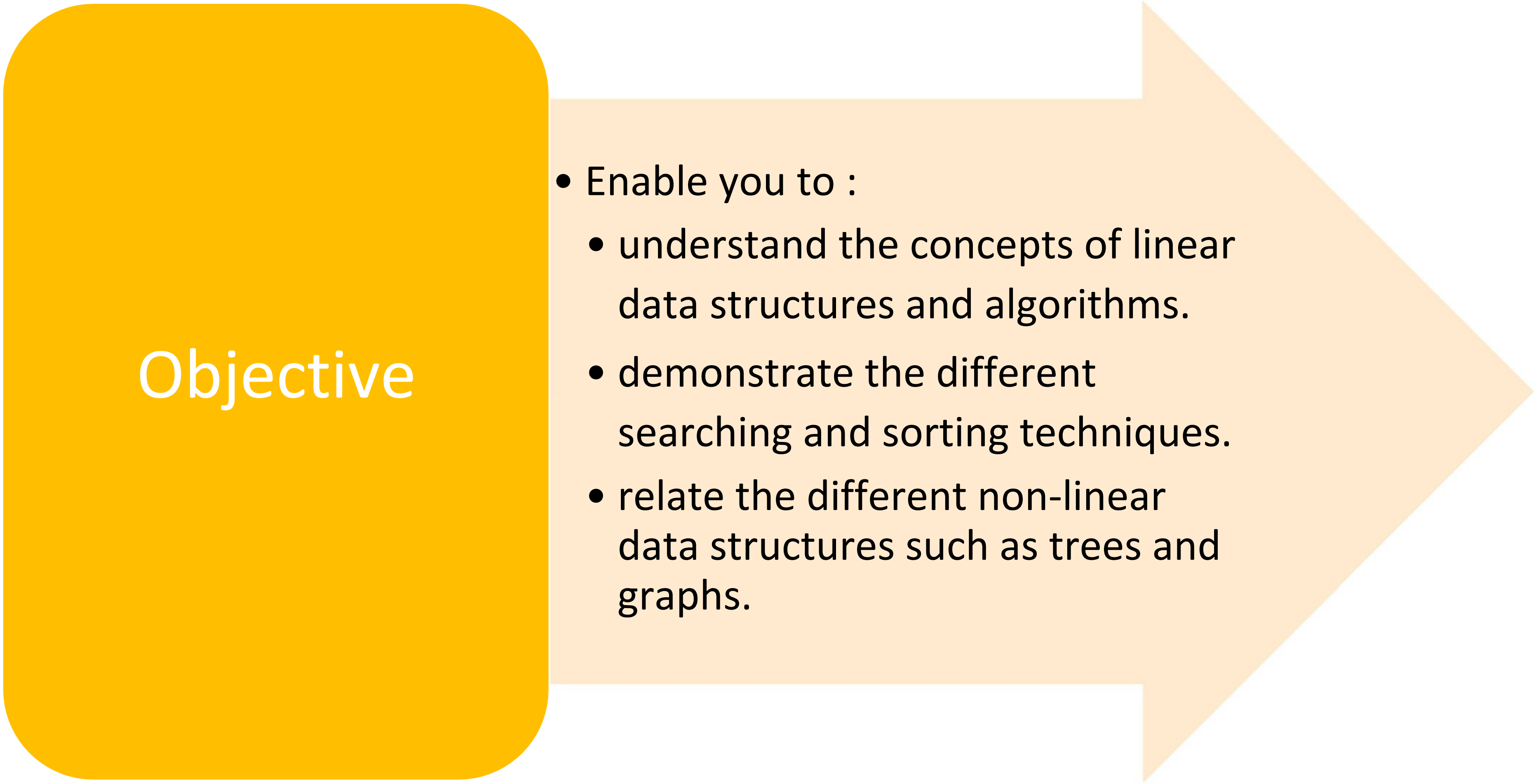


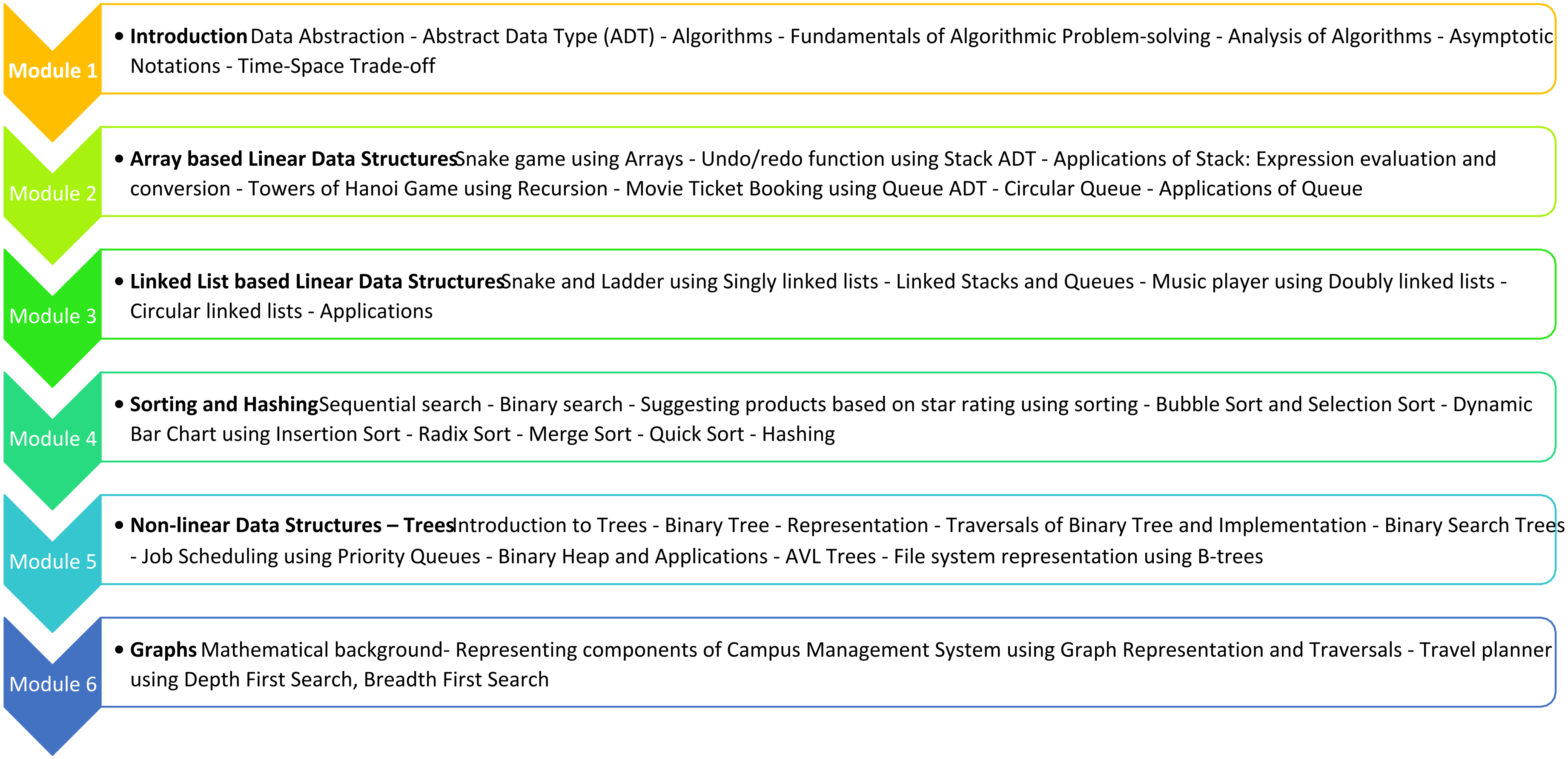
20CS2013 - DATA STRUCTURES AND ALGORITHMS



Objective

- Enable you to :
  - understand the concepts of linear data structures and algorithms.
  - demonstrate the different searching and sorting techniques.
  - relate the different non-linear data structures such as trees and graphs.

SYLLABU



Module 1

- **Introduction** Data Abstraction - Abstract Data Type (ADT) - Algorithms - Fundamentals of Algorithmic Problem-solving - Analysis of Algorithms - Asymptotic Notations - Time-Space Trade-off

Module 2

- **Array based Linear Data Structures** Snake game using Arrays - Undo/redo function using Stack ADT - Applications of Stack: Expression evaluation and conversion - Towers of Hanoi Game using Recursion - Movie Ticket Booking using Queue ADT - Circular Queue - Applications of Queue

Module 3

- **Linked List based Linear Data Structures** Snake and Ladder using Singly linked lists - Linked Stacks and Queues - Music player using Doubly linked lists - Circular linked lists - Applications

Module 4

- **Sorting and Hashing** Sequential search - Binary search - Suggesting products based on star rating using sorting - Bubble Sort and Selection Sort - Dynamic Bar Chart using Insertion Sort - Radix Sort - Merge Sort - Quick Sort - Hashing

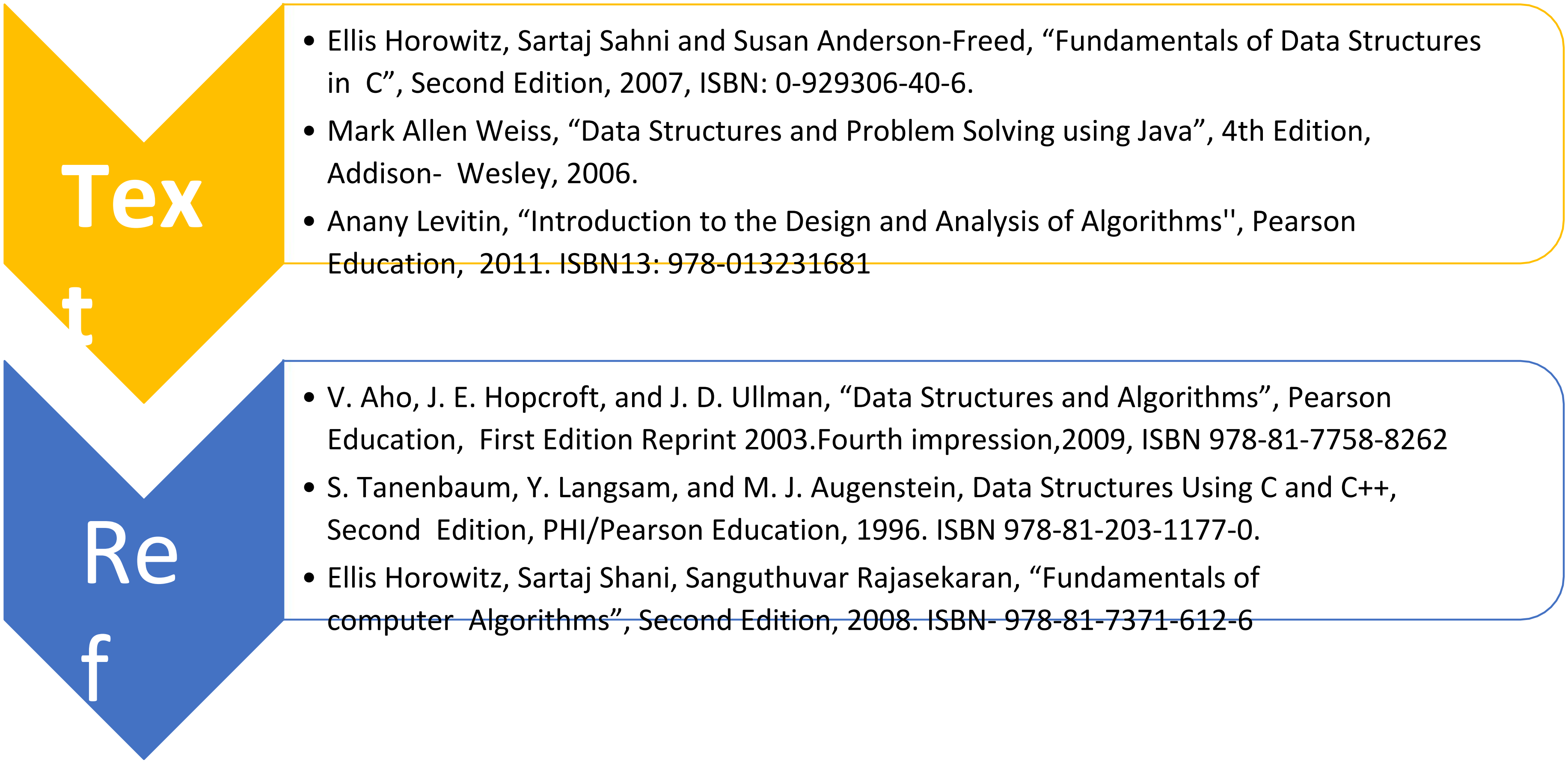
Module 5

- **Non-linear Data Structures – Trees** Introduction to Trees - Binary Tree - Representation - Traversals of Binary Tree and Implementation - Binary Search Trees - Job Scheduling using Priority Queues - Binary Heap and Applications - AVL Trees - File system representation using B-trees

Module 6

- **Graphs** Mathematical background- Representing components of Campus Management System using Graph Representation and Traversals - Travel planner using Depth First Search, Breadth First Search

REFERENC



Text

- Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, “Fundamentals of Data Structures in C”, Second Edition, 2007, ISBN: 0-929306-40-6.
- Mark Allen Weiss, “Data Structures and Problem Solving using Java”, 4th Edition, Addison- Wesley, 2006.
- Anany Levitin, “Introduction to the Design and Analysis of Algorithms'', Pearson Education, 2011. ISBN13: 978-013231681

Ref

- V. Aho, J. E. Hopcroft, and J. D. Ullman, “Data Structures and Algorithms”, Pearson Education, First Edition Reprint 2003.Fourth impression,2009, ISBN 978-81-7758-8262
- S. Tanenbaum, Y. Langsam, and M. J. Augenstein, Data Structures Using C and C++, Second Edition, PHI/Pearson Education, 1996. ISBN 978-81-203-1177-0.
- Ellis Horowitz, Sartaj Shani, Sanguthuvar Rajasekaran, “Fundamentals of computer Algorithms”, Second Edition, 2008. ISBN- 978-81-7371-612-6

Outcome

- You will be able to :
  - understand the basics of **abstract data type** and **algorithm analysis**
  - illustrate the use of array to implement **stack and queue**
  - apply **linked list** to design stack and queue data structures
  - demonstrate the working of **sorting and searching** algorithms and application of **hashing**
  - understand the different types of **tree data structures** and demonstrate the methods for **traversing trees**
  - differentiate the **graph representations** and traversals

Good Computer Program

- A computer program is a series of instructions to carry out a particular task written in a language that a computer can understand.
- The process of preparing and feeding the instructions into the computer for execution is referred as programming.
- There are a number of features for a good program

  Run efficiently and correctly

  Have a user friendly interface

  Be easy to read and understand

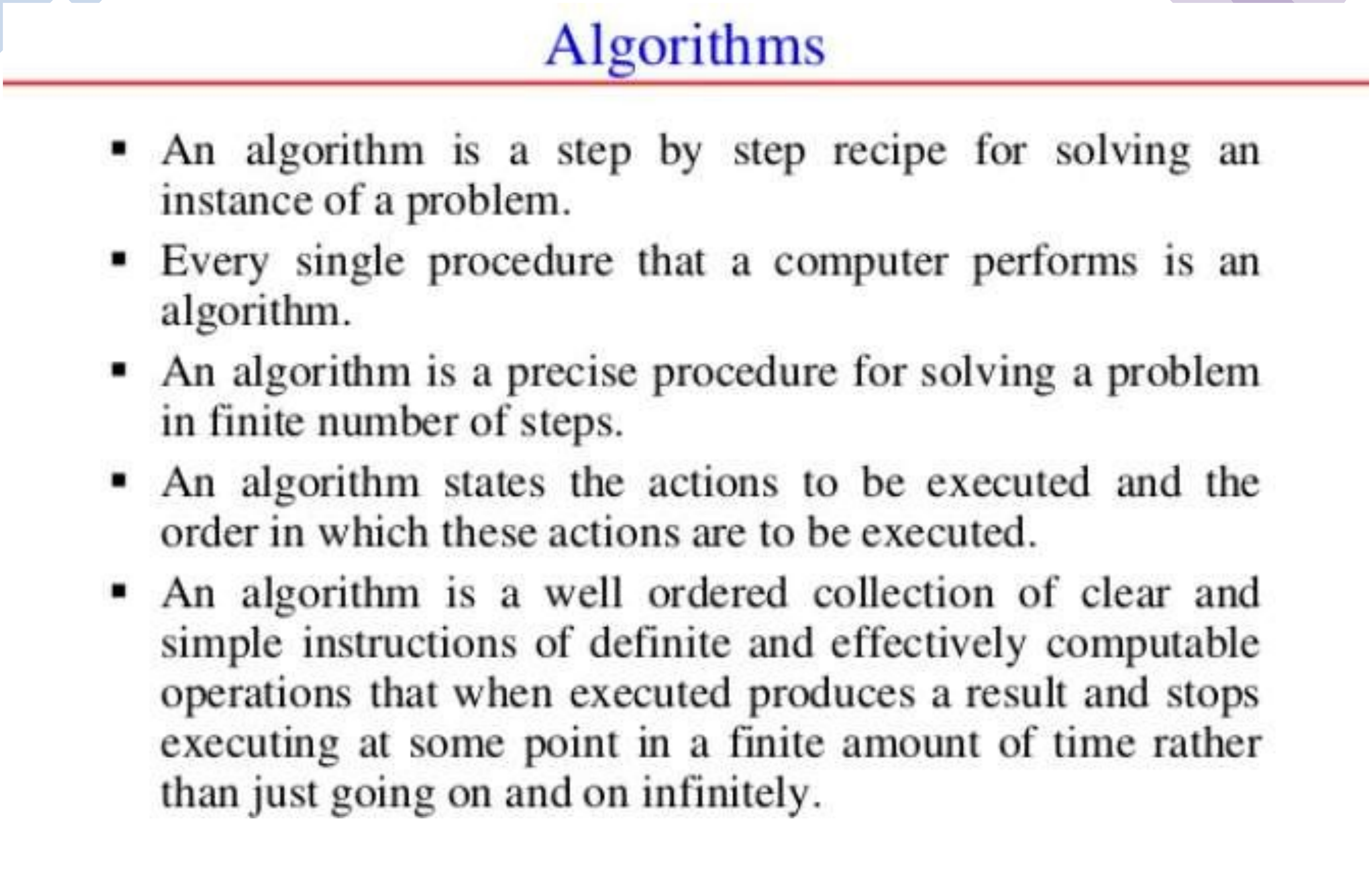
  Be easy to debug

  Be easy to modify

  Be easy to maintain

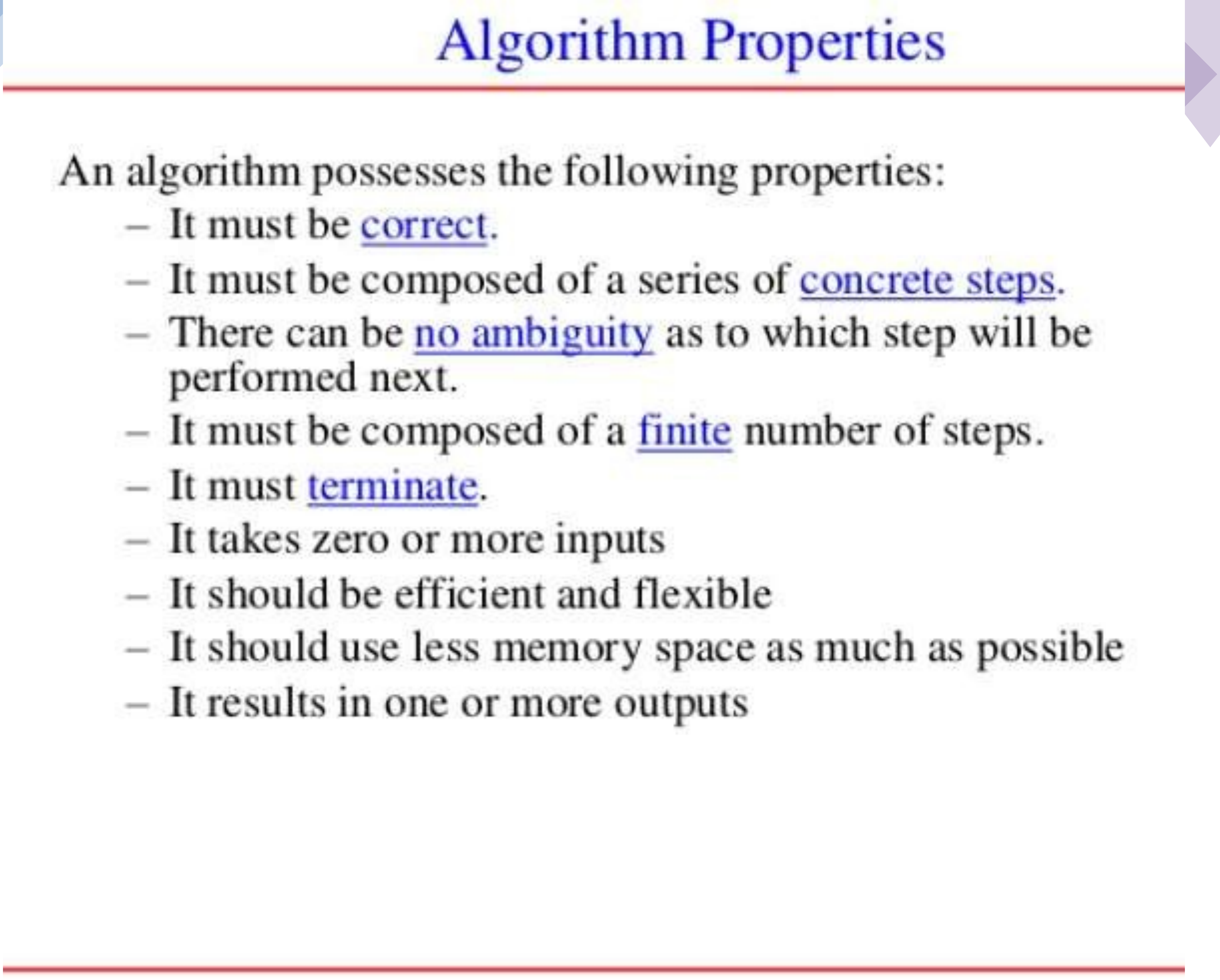
Good Computer Program

- Programs consists of two things: Algorithms and data structures
- A Good Program is a combination of both algorithm and a data structure
- An algorithm is a step by step recipe for solving an instance of a problem
- A data structure represents the logical relationship that exists between individual elements of data to carry out certain tasks
- A data structure defines a way of organizing all data items that consider not only the elements stored but also stores the relationship between the elements



Algorithms

- An algorithm is a step by step recipe for solving an instance of a problem.
- Every single procedure that a computer performs is an algorithm.
- An algorithm is a precise procedure for solving a problem in finite number of steps.
- An algorithm states the actions to be executed and the order in which these actions are to be executed.
- An algorithm is a well ordered collection of clear and simple instructions of definite and effectively computable operations that when executed produces a result and stops executing at some point in a finite amount of time rather than just going on and on infinitely.



Algorithm Properties

An algorithm possesses the following properties:

- It must be correct.
- It must be composed of a series of concrete steps.
- There can be no ambiguity as to which step will be performed next.
- It must be composed of a finite number of steps.
- It must terminate.
- It takes zero or more inputs
- It should be efficient and flexible
- It should use less memory space as much as possible
- It results in one or more outputs

The Need for Data Structures

Data structures organize data

⇒ more efficient programs.

More powerful computers ⇒ more complex applications.

More complex applications demand more calculations.

Complex computing tasks are unlike our everyday experience.

- More typically, a *data structure* is meant to be an *organization for a collection of data items*.
- Any organization for a collection of records can be searched, processed in any order, or modified.
- The choice of data structure and algorithm can make the difference between a program running in a few seconds or many days. A data structure requires a certain amount of:
- space for each data item it stores
- time to perform a single basic operation
- programming effort.

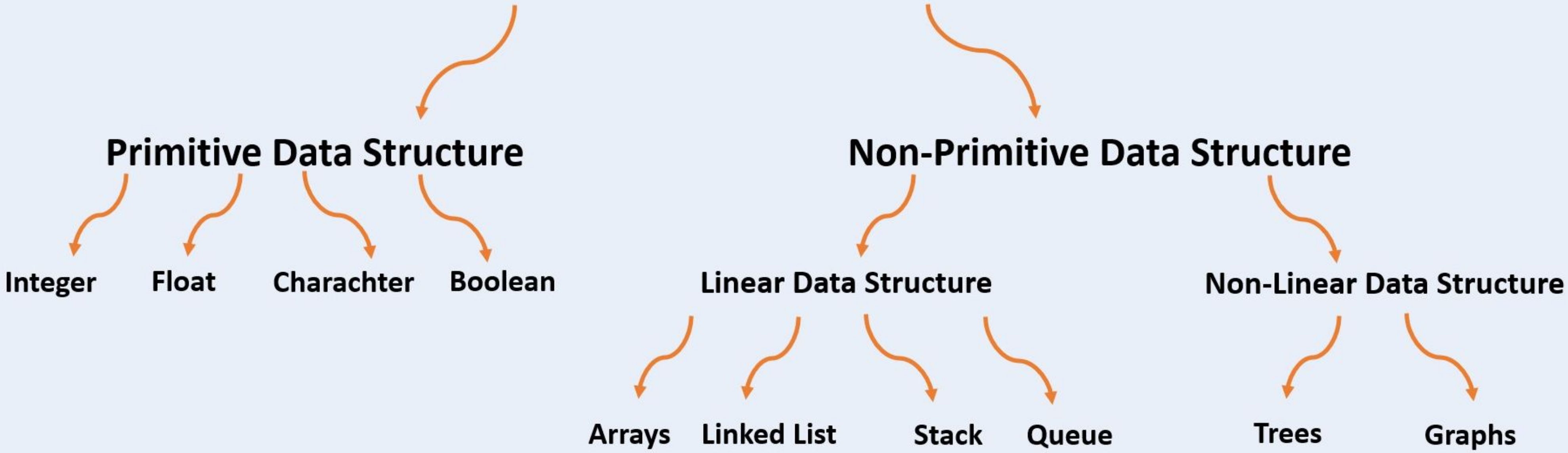
Selecting a Data Structure

Select a data structure as follows:

1. Analyze the problem to determine the resource constraints a solution must meet.
2. Determine the basic operations that must be supported. Quantify the resource constraints for each operation.
3. Select the data structure that best meets these requirements.

“*Data Structures* ”deals with the study of how the data is organized in the memory, how efficiently the data can be retrieved and manipulated, and the possible ways in which different data items are logically related.

Types of Data Structure



Primitive Data Structure

Integer

Float

Charachter

Boolean

Non-Primitive Data Structure

Linear Data Structure

Arrays

Linked List

Stack

Queue

Non-Linear Data Structure

Trees

Graphs

THANK YOU