TECHNO INDIA UNIVERSITY
WESTBENGAL

Data Structures & Algorithms
(PCC-CS 301)

Dr. Debashis Das
Associate Professor
Department of CSE
Techno India University, Kolkata

Topics Covered

1. Introduction to Data Structures
2. Introduction to Algorithms
3. Overview of the Course
4. Books to be followed

Introduction to Data Structures

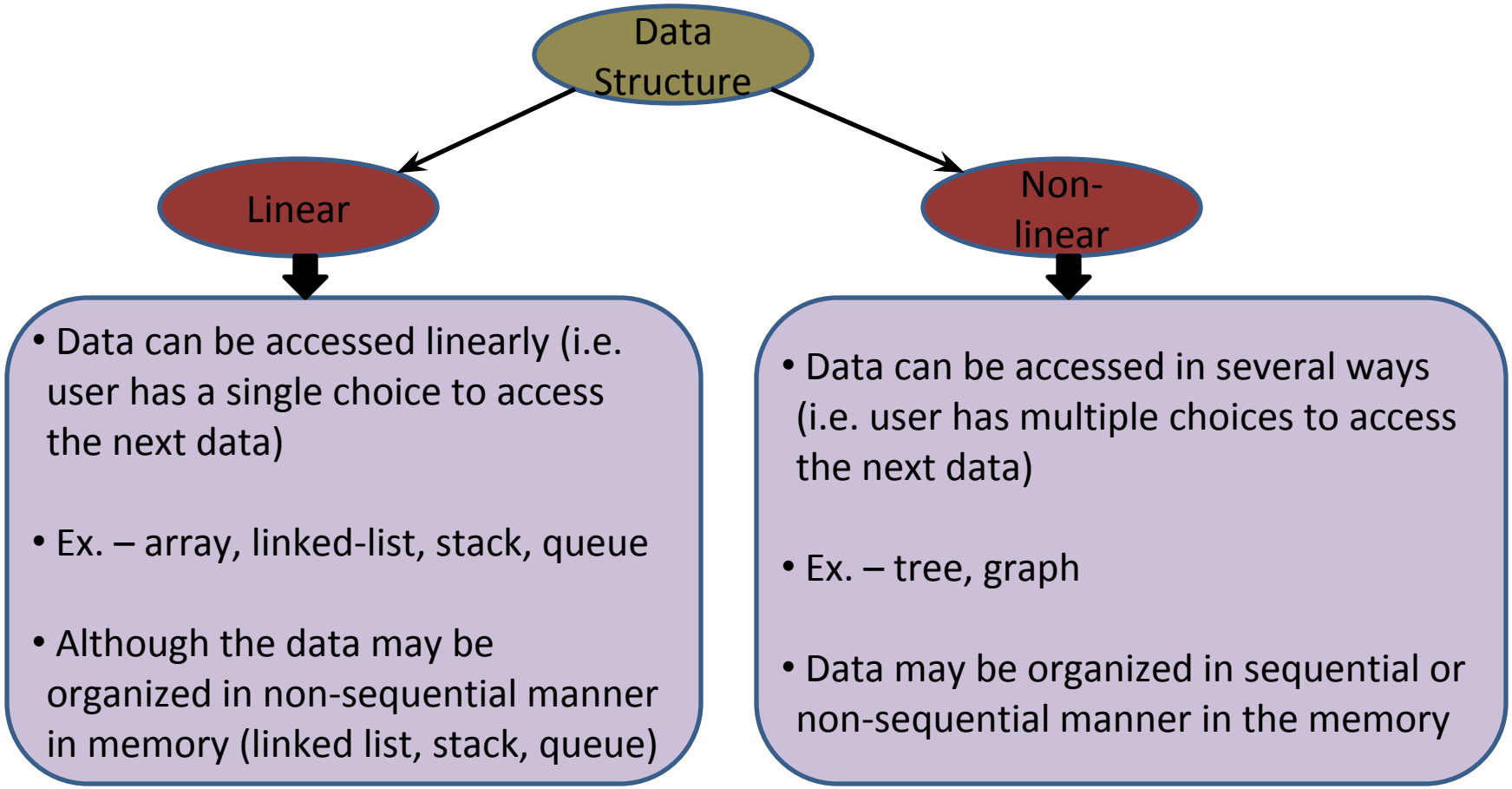
- Data Structure:

  - Definition
    - Data structure is a way of **storing data** in such an **organized manner** so that it can be **accessed efficiently**

  - Need
    - To store a single data in computer memory, it is sufficient to use a variable of primitive type (int, float, char etc.)
    - To store a large number of related data (similar/dissimilar type), we need to use a data structure

Introduction to Data Structures

- Data Structure: classification



Introduction to Data Structures

- Abstract Data Type (ADT):

  - Motivation
    - To solve any problem, we need to use certain data structures and a set of operations that to be performed on those data structures
    - To simplify the process of solving the problems, the data structures are sometimes combined with their operations which is called **abstract data type**
  - Example:
    - Linked-list, stack, queue, priority queue, binary tree, dictionary, disjoint set, hash table, graph etc.

Introduction to Data Structures

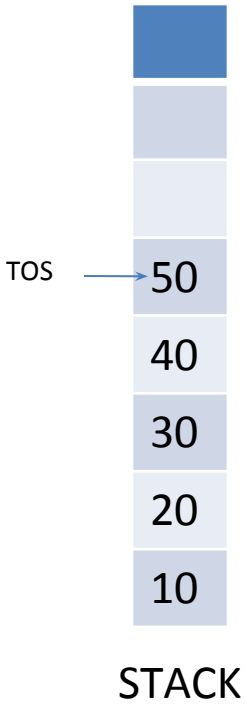
- Abstract Data Type (ADT):

  - Components
    - Declaration of data structure
    - Declaration of operations

  - Properties
    - Implementation of an ADT remains hidden (or abstract), user can implement it by their own way
    - Only the operations and associated constraints on the data structures are defined

Introduction to Data Structures

- Abstract Data Type (ADT): example
  - Stack
    - Associated operations
      - PUSH (inserting an element into stack)
      - POP (deleting the top most element of the stack)
      - DISPLAY (showing the elements from top of the stack)
    - Constraints
      - It follows Last In First Out (LIFO), an element inserted last will be deleted first
      - Only the Top most element can be accessible through a pointer called top of the stack (TOS)
      - Stack should be of fixed size

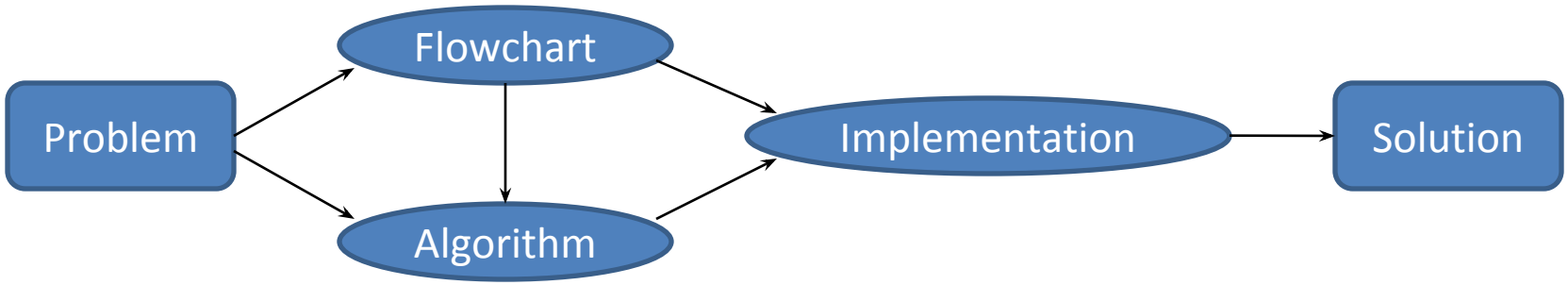


Introduction to Algorithms

- Algorithm

  - Definition
    - An algorithm is a well defined set of instructions to solve a given problem

  - Problem Solving Steps



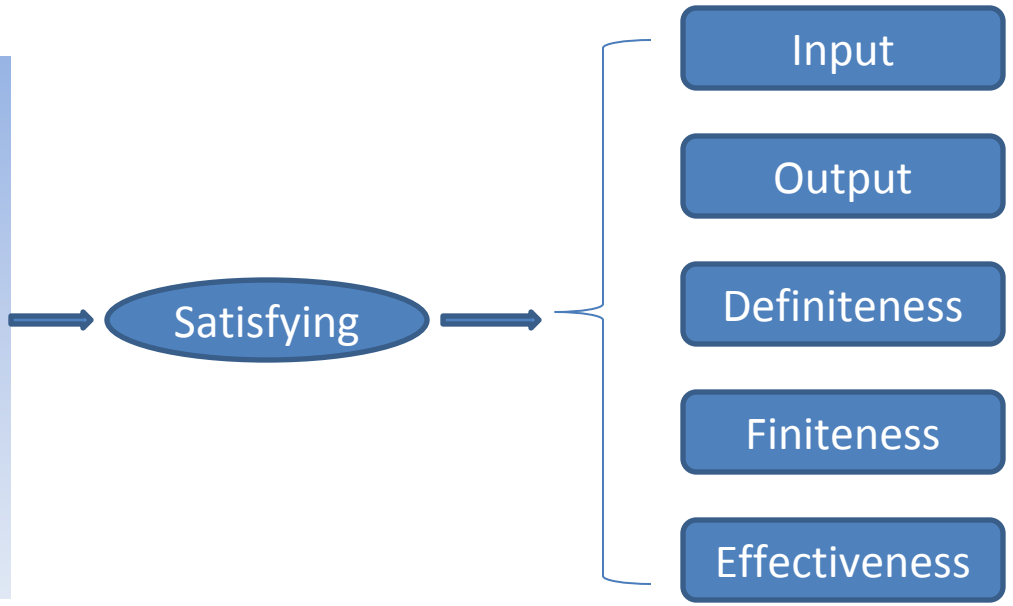
Introduction to Algorithms

- Algorithm: properties
  - Input
    - There should be zero or more inputs into an algorithm
  - Output
    - It should produce at least one output
  - Definiteness
    - Each instructions must be clear and unambiguous
  - Finiteness
    - For every input it should be stopped after a finite amount of time
  - Effectiveness
    - All the instructions should be feasible (implementable)

Introduction to Algorithms

- Can you write an algorithm for any problem
  - A given number is odd or even?

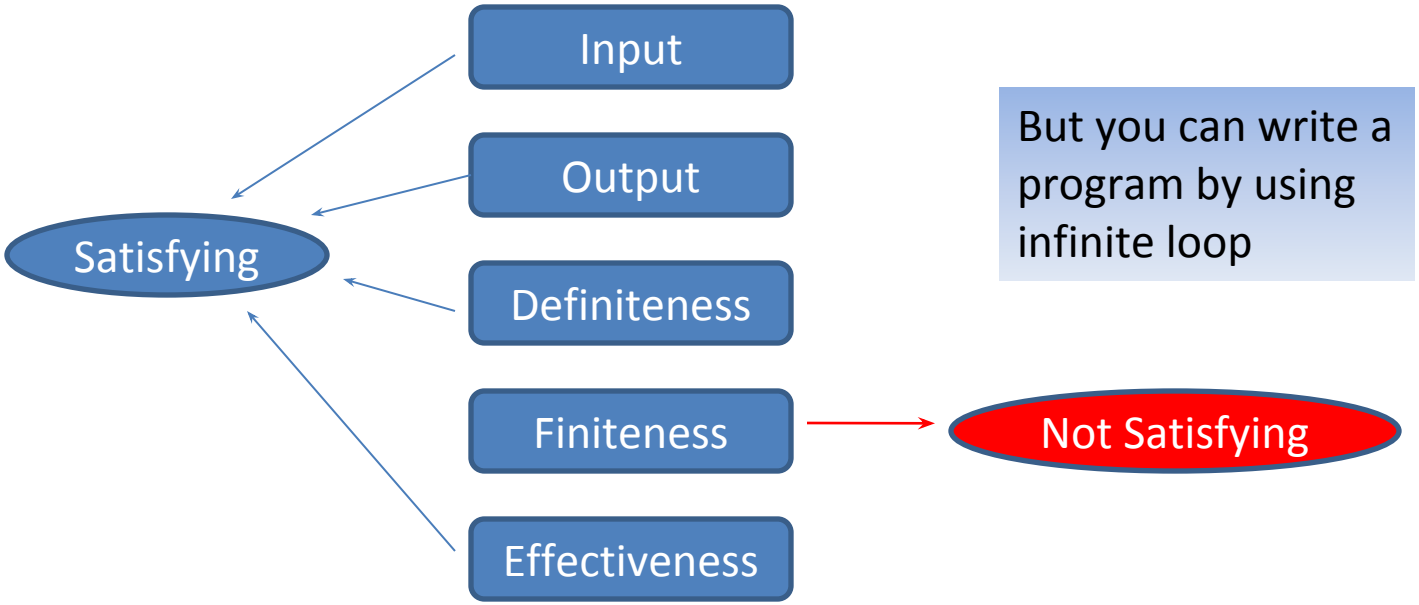
```
Odd_even_test ( n )
{
  m := n (mod) 2
  If m := 0
     Print n as EVEN
  Else
     Print n as ODD
  return
}
```



Algorithm is possible as all the criteria fulfil

Introduction to Algorithms

- Can you write an algorithm for any problem
  - Find and print all prime numbers.



But you can write a program by using infinite loop

Algorithm is not possible as all the criteria are not satisfying

Introduction to Algorithms

- Algorithm: performance analysis

  - Motivation
    - To solve a problem there may have several algorithms
    - Most efficient and suitable algorithm will be in demand
    - Thus the performance analysis is necessary

  - Comparison measure
    - Time complexity
    - Space complexity

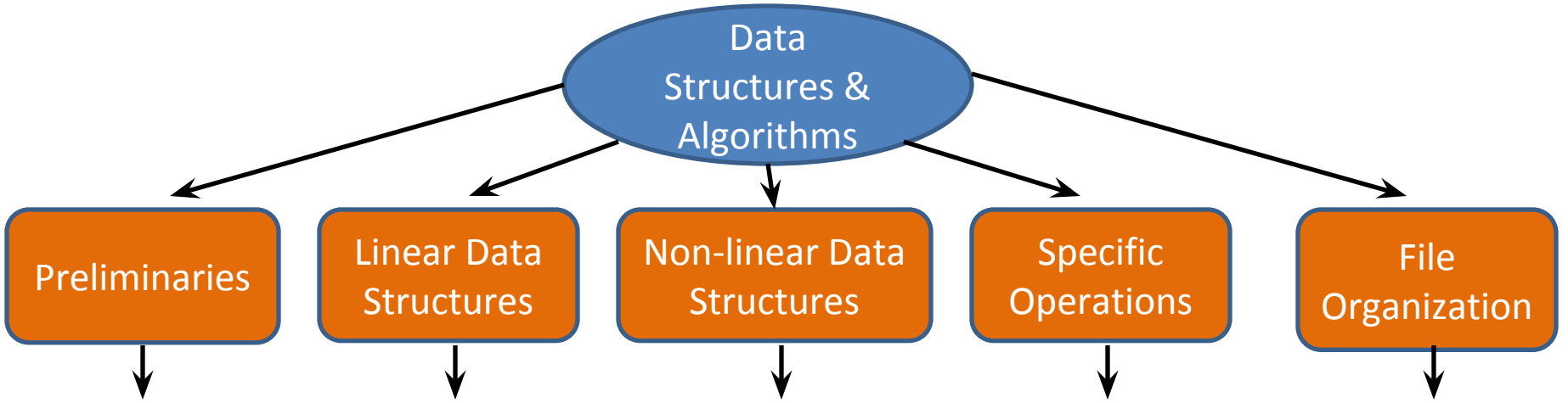
Introduction to Algorithms

- Algorithm: performance analysis

  - Time complexity
    - It is measured as the running time of an algorithm as a function of the input size
    - The rate at which the running time increases with the input size (called as rate of growth)
    - It is **not measured** by the time taken in minute/sec/millisecond because an instruction execution time is purely architecture dependent

  - Space complexity
    - It is measured through the amount of primary memory consumes by the algorithm during implementation

Overview of the Course



Introduction to data structures & algorithms, performance analysis, ADT, recursion, asymptotic notation, time-space trade-off

Array, Linked list, Stack, Queue and it's variants, it's different representations, associated operations, applications

Tree and it's variants, Graph and it's variants, memory representations, applications

Searching: linear & binary search, DFS, BFS, hashing
Sorting: Comparison based sorting, divide-conquer based sorting

Organization, different types of accessing schemes

Books and References

- Text Book:
  - ✔ *Fundamentals of Data Structures, E. Horowitz and S. Sahni*
  - ✔ *Data Structures and Algorithms, Alfred V. Aho, John E. Hoppcroft, Jeffrey D. Ullman*

- Reference Book:
  - ✔ *The Art of Computer programming: Volume I: Fundamental Algorithms, Donald E. Knuth*
  - ✔ *Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein*
  - ✔ *Open Data Structures: An Introduction (Open paths to Enriched Learning), 31st Edition, Pat Morin*
  - ✔ *Data Structures and Algorithms Made Easy, N. Karumanchi*

Queries?