Introduction of Data Structures and Algorithms

Definition

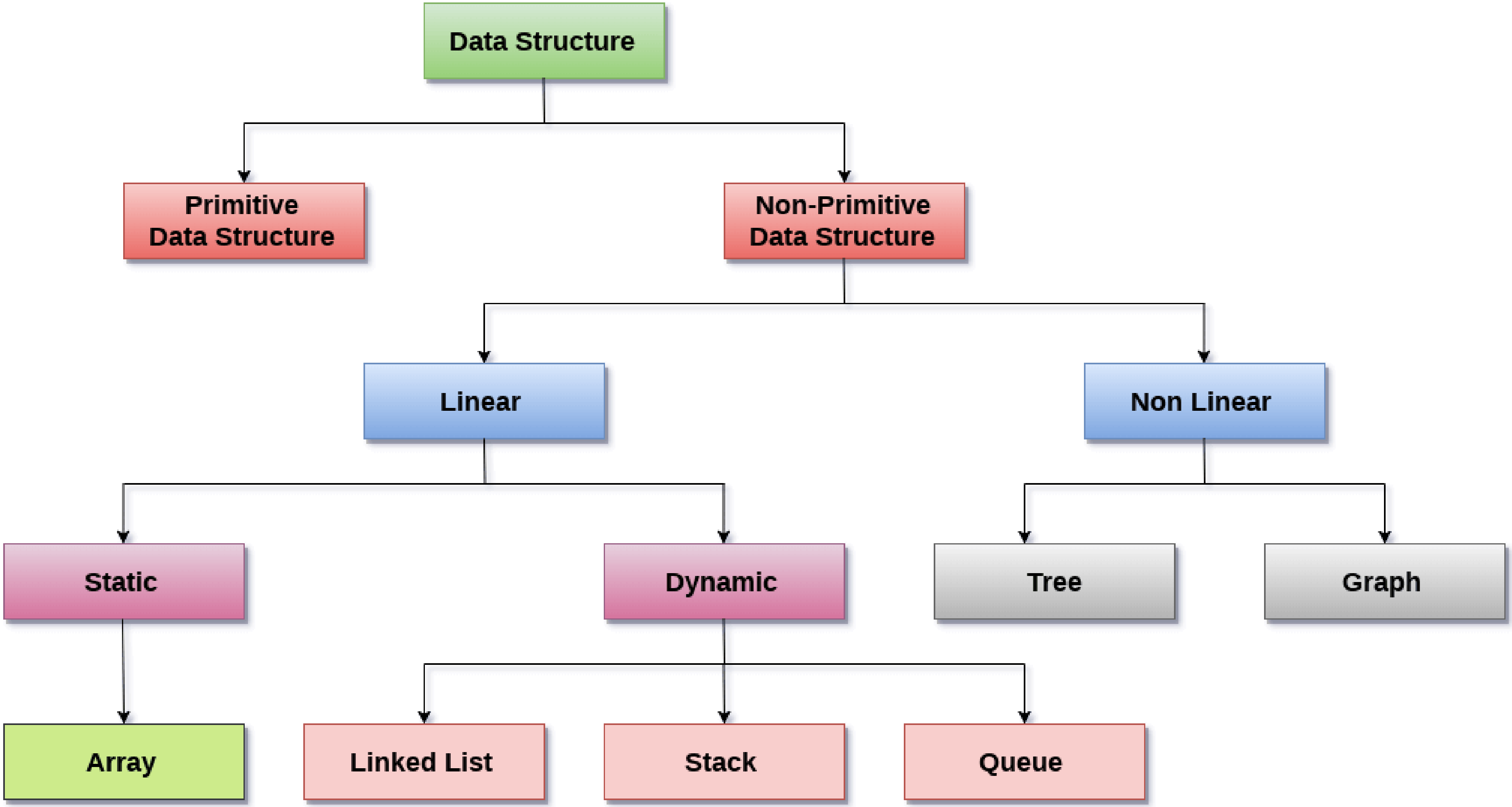
- Data structure is representation of the logical relationship existing between individual elements of data.
- In other words, a data structure is a way of organizing all data items that considers not only the elements stored but also their relationship to each other.
- Data structure affects the design of both structural and functional aspects of a program.

Program= Algorithm + Data Structure

Introduction

- Algorithm is a set of instruction written to carry out certain tasks & the data structure is the way of organizing the data with their logical relationship retained.
- To develop a program of an algorithm, we should select an appropriate data structure for that algorithm.
- Therefore algorithm and its associated data structures from a program.

Classification of Data Structure



Data structures operations

The most commonly used operation on data structure are broadly categorized into following types:

- Create
- Selection
- Updating
- Searching
- Sorting
- Merging
- Delete

Data Structures : Arrays

- An array is defined as a set of finite number of homogeneous elements or same data items.
- It means an array can contain one type of data only, either all integer, all float-point number or all character.

Arrays

- Insertion of new element
- Deletion of required element
- Modification of an element
- Merging of arrays

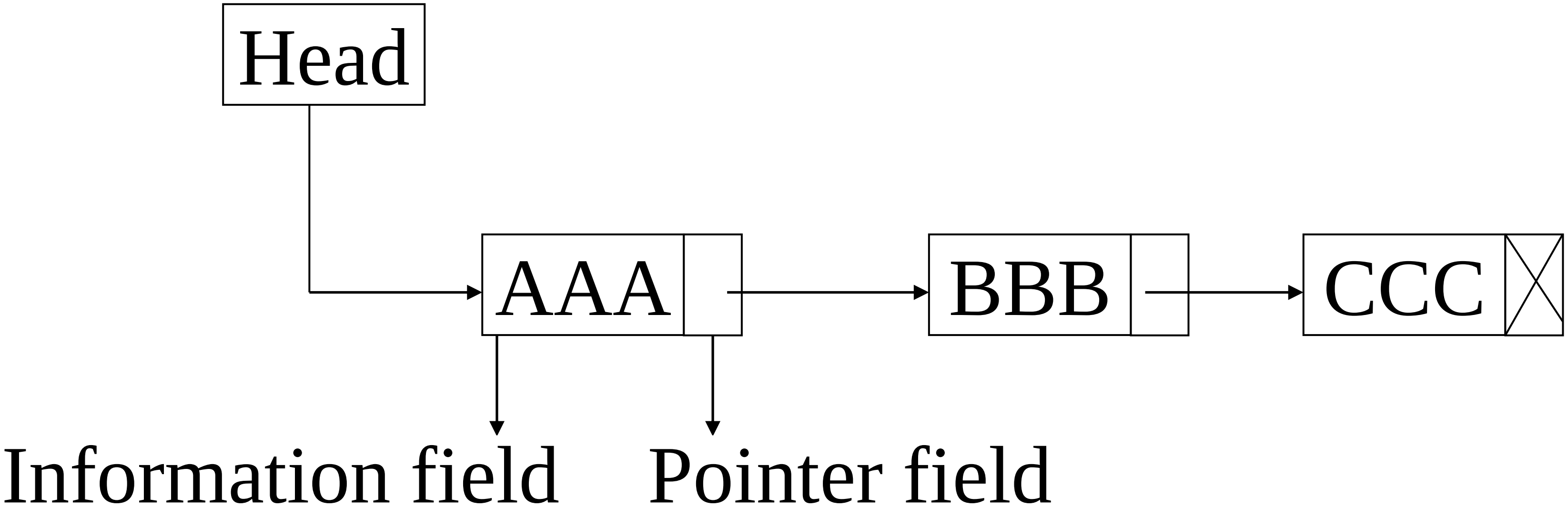
Linked Lists

- A lists (Linear linked list) can be defined as a collection of variable number of data items.
- Lists are the most commonly used non-primitive data structures.
- An element of list must contain at least two fields, one for storing data or information and other for storing address of next element.
- As you know for storing address we have a special data structure of list the address must be pointer type.

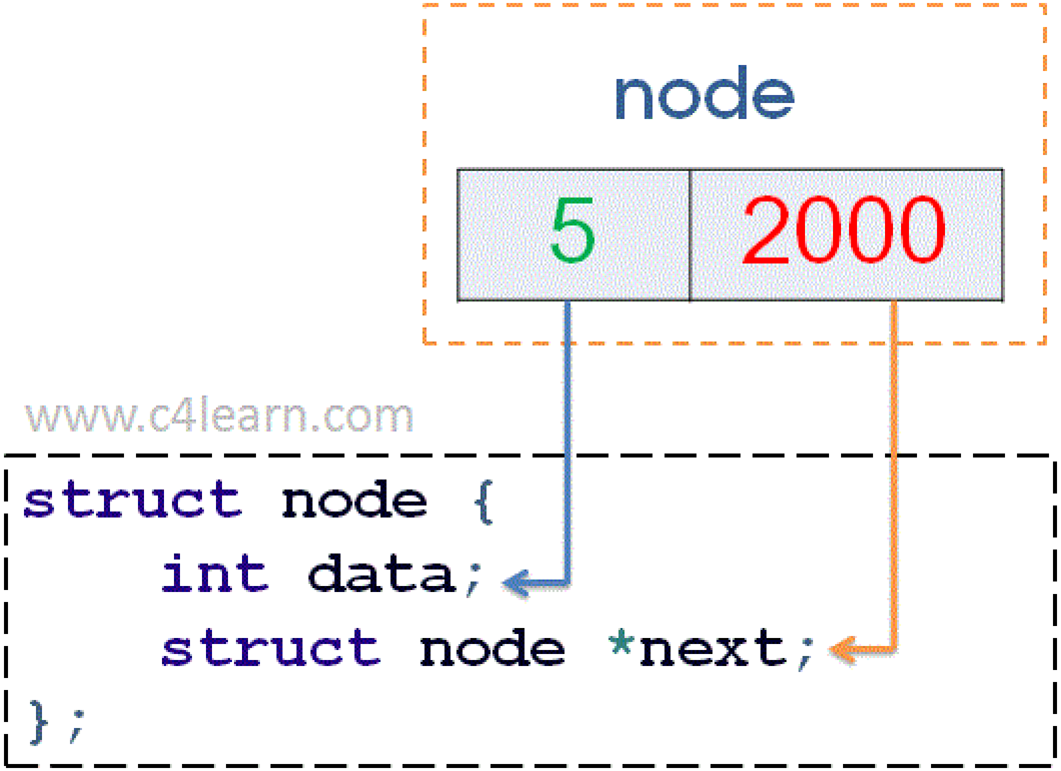
Lists

- Technically each such element is referred to as a node, therefore a list can be defined as a collection of nodes as show bellow:

[Linear Liked List]



Node Structure



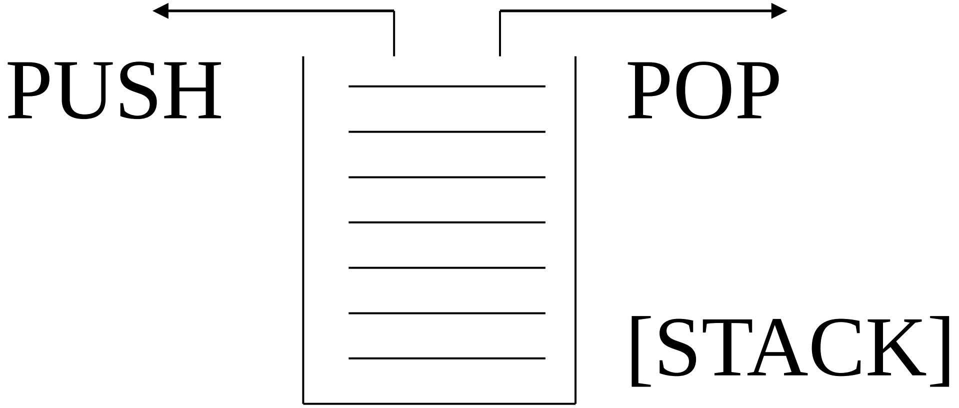
```
struct node {
     int data;
     struct node *next;
};
```

Stack

- A stack is also an ordered collection of elements like arrays, but it has a special feature that deletion and insertion of elements can be done only from one end called the top of the stack (TOP)
- Due to this property it is also called as last in first out type of data structure (LIFO).

Stack

- Insertion of element into stack is called PUSH and deletion of element from stack is called POP.
- The bellow show figure how the operations take place on a stack:



Stack

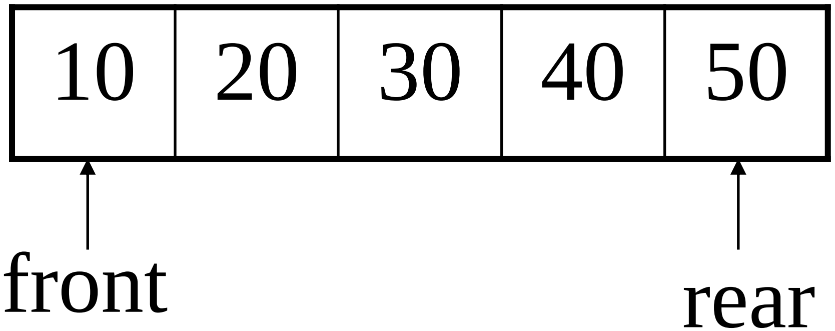
- The stack can be implemented into two ways:
  - Using arrays (Static implementation)
  - Using pointer (Dynamic implementation)

Queue

- Queue are first in first out type of data structure (i.e. FIFO)
- In a queue new elements are added to the queue from one end called REAR end and the element are always removed from other end called the FRONT end.
- The people standing in a railway reservation row are an example of queue.

Queue

- Each new person comes and stands at the end of the row and person getting their reservation confirmed get out of the row from the front end.
- The bellow show figure how the operations take place on a queue:



10	20	30	40	50

front

rear

Queue

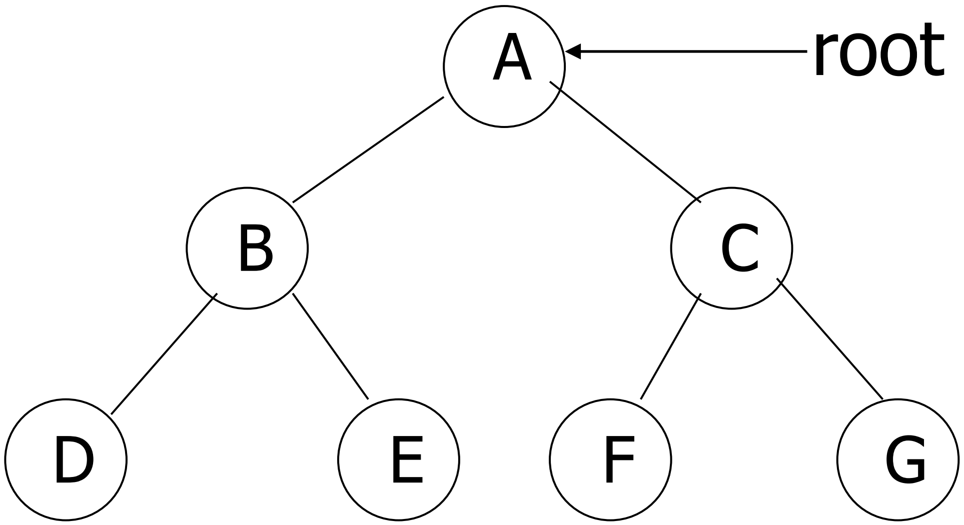
- The queue can be implemented into two ways:
    - Using arrays (Static implementation)
    - Using pointer (Dynamic implementation)

Trees

- The tree is a nonlinear hierarchical data structure and comprises a collection of entities known as nodes. It connects each node in the tree data structure using "edges”, both directed and undirected.
- There is a special data item at the top of hierarchy called the Root of the tree.
- The remaining data items are partitioned into number of mutually exclusive subset, each of which is itself, a tree which is called the sub tree.
- The tree always grows in length towards bottom in data structures, unlike natural trees which grows upwards.

Trees

- The tree structure organizes the data into branches, which related the information.

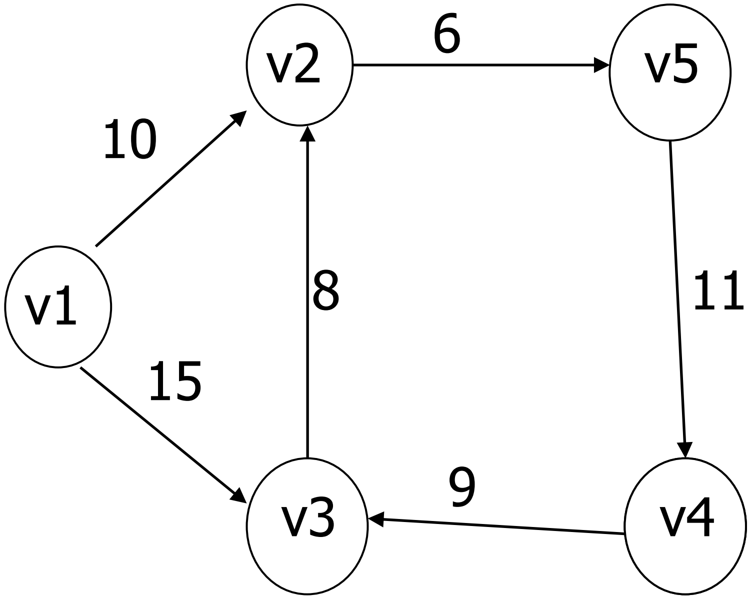


Graph

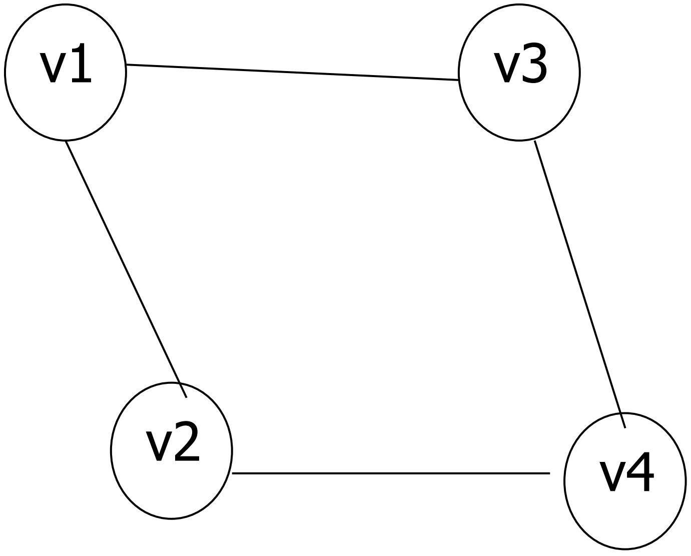
- Graph is a non-linear data structure capable of representing many kind of physical structures.
- It has found application in Geography, Chemistry and Engineering sciences.
- Definition: A graph G(V,E) is a set of vertices V and a set of edges E.
- An edge connects a pair of vertices and many have weight such as length, cost and another measuring instrument for according the graph.
- Vertices on the graph are shown as point or circles and edges are drawn as arcs or line segment.

Graph

- Example of graph:



[a] Directed & Weighted Graph



[b] Undirected Graph

Graph

- Types of Graphs:
    - Directed graph
    - Undirected graph
    - Weighted graph
    - Connected graph
    - Non-connected graph

Real-life applications

- You have to store social network “feeds”. You do not know the size, and things may need to be dynamically added.
- You need to store undo/redo operations in a word processor.
- You need to evaluate an expression.
- You need to store the friendship information on a social networking site. i.e., who is friends with who.
- You need to store an image (1000 by 1000 pixels) as a bitmap.

- To implement printer spooler so that jobs can be printed in the order of their arrival.
- To implement back functionality in the internet browser.
- To store a set of fixed key words which are referenced very frequently.
- To store the customer order information in a drive-in burger place. (Customers keep on coming and they have to get their correct food at the payment/food collection window.)
- To store the genealogy information of biological species.