Linked Lists

Unit: 2

Data Structures

Unit II Syllabus

- Advantages of linked list over array,
- Self-referential structure,
- Singly Linked List, Doubly Linked List, Circular Linked List.
- **Operations on a Linked List:** Insertion, Deletion, Traversal, Reversal, Searching, Polynomial Representation and Addition of Polynomials.
- Implementation of Stack and Queue using Linked lists.

Unit Content

- Advantages of Linked List over Array
- Singly Linked List
- Doubly Linked List
- Circular Linked List
- Circular doubly linked list
- Operation on Linked List
  - Insertion
  - Deletion
  - Traversal
  - Reversal
  - Searching Polynomial Representation
  - Addition, Subtraction and Multiplication of Polynomials
- Implementation of Stack and Queue using Linked List

Unit Objective

- To learn about linked lists.
- To understand different types of Linked list.
- Basic operations of linked list.

Course Objective

- Introduction to basic data structures.
- To know about the basic properties of different data structures.
- Classification and operations on data structure
- Understand algorithms and their efficiency
- Study logical and mathematical description of array and link list.
- Implementation of array and link list on computer.
- Differentiate the usage of array and link list in different scenarios.

Faculty Video Links, Youtube & NPTEL Video Links and Online Courses Details

- Youtube/other Video Links
- Implementation of link list
  - https://www.youtube.com/watch?v=6wXZ_m3SbEs
- Polynomial addition using link list
  - https://www.youtube.com/watch?v=V_ZNKu_pUPQ

Basic Terminology(CO1)

- **Linked List**
- **Doubly Linked List**
- **Circularly Linked List**
- **Circularly Doubly Linked List**

Topic Objective

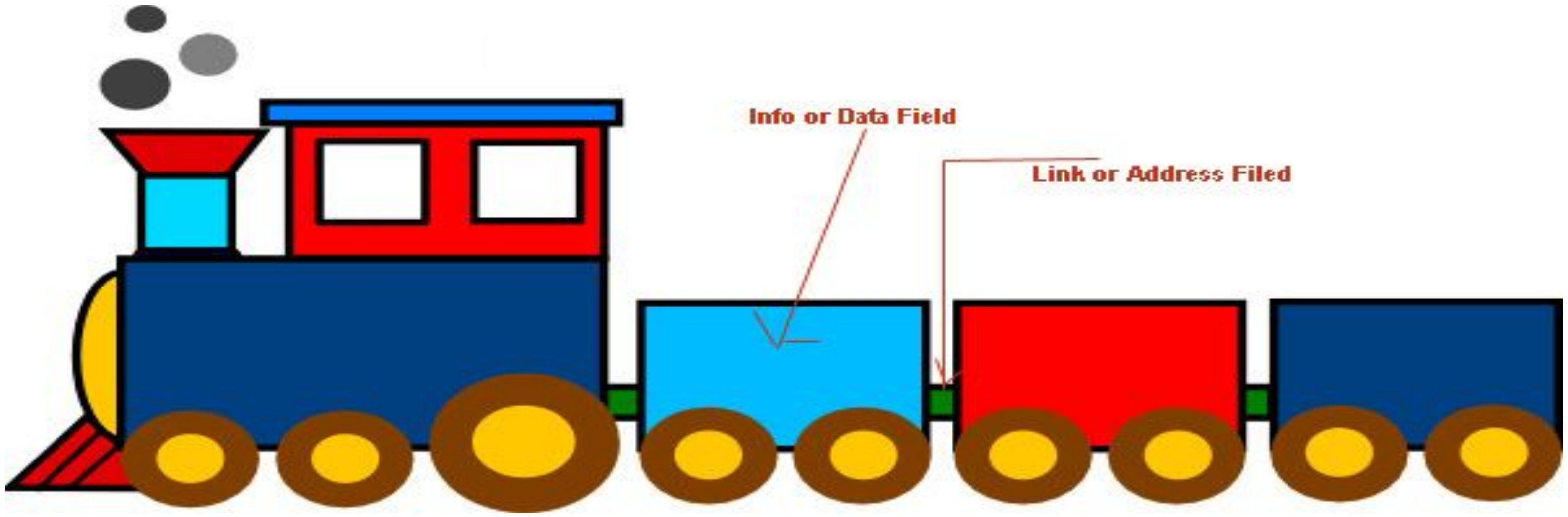
- To understand linked list and the operations of linked list.
- To implement Linked list program using Python

Linked List

- Linked List can be defined as collection of objects called nodes that are randomly stored in the memory.
- A node contains two fields i.e. data stored at that particular address and the pointer which contains the address of the next node in the memory.
- The last node of the list contains pointer to the null.

Linked List

- A linked list is a linear data structure.
- Nodes make up linked lists.
- Nodes are structures made up of data and a pointer to another node.
- Usually the pointer is called next.



Introduction to Linked List

Linked
List

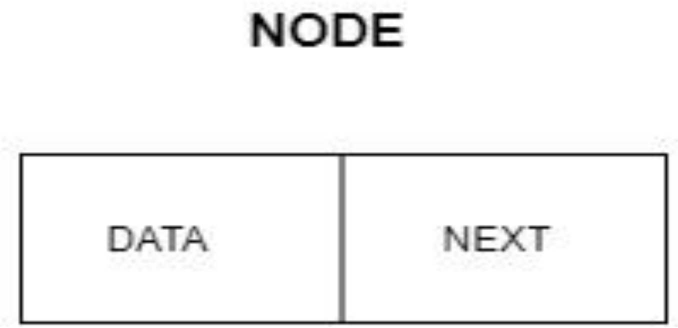
- **The elements of a linked list are not stored in adjacent memory locations as in arrays.**
- **It is a linear collection of data elements, called nodes.**

Continued….

Linked List

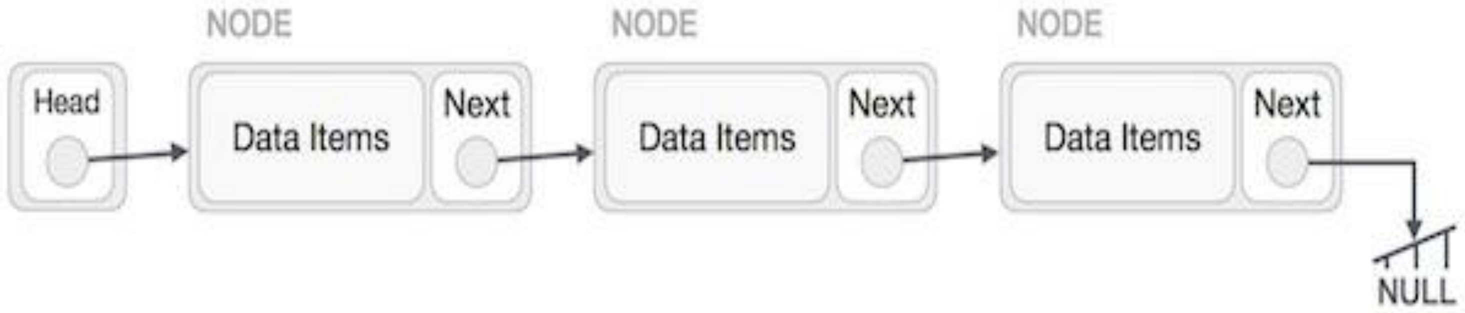
- **In a linear or single-linked list, a node is connected to the next node by a single link.**

- **A node in this type of linked list contains two types of fields**
  - **data: which holds a list element**
  - **next: which stores a link (i.e. pointer) to the next node in the list.**



Linked List Representation

- Linked list can be visualized as a chain of nodes, where every node points to the next node.



- As per the above illustration, following are the important points to be considered.
  - Linked List contains a link element called first(Head).
  - Each link carries a data field(s) and a link field called next.
  - Each link is linked with its next link using its next link.
  - Last link carries a link as null to mark the end of the list.

Properties of linked list

- The nodes in a linked list are not stored contiguously in the memory
- You don’t have to shift any element in the list
- Memory for each node can be allocated dynamically whenever the need arises.
- The size of a linked list can grow or shrink dynamically