Chapter 2.

LINKED LIST

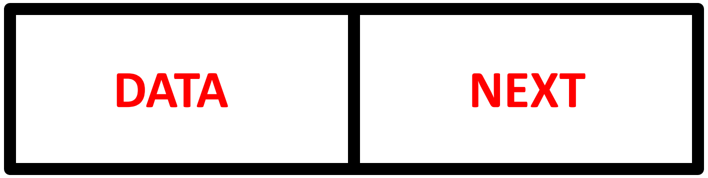
Introduction

Defination

- "Linked List is order collection of data in which each element (node) contains the data and link to its successor."
- A node contains two fields

    1. data

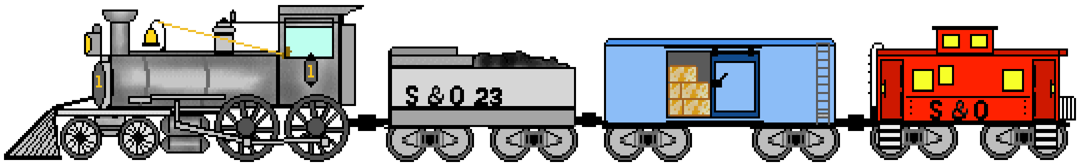
    2. pointer which contains the address o the next node



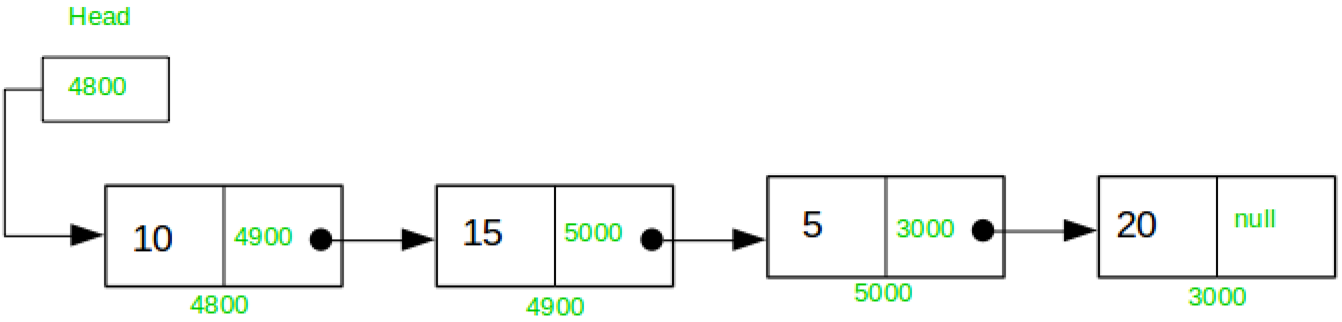
DATA NEXT

Node

Real Life Example



Singly Linked list



Head

4800

10 | 4900 → 4800

15 | 5000 → 4900

5 | 3000 → 5000

20 | null → 3000

Advantage of Link List

- Linked List is Dynamic data **Structure** .
- Linked List can grow and shrink during run time.
- Insertion and Deletion Operations are Easier.
- Efficient **Memory** Utilization ,i.e no need to pre-allocate **memory**.
- Many complex application can be easily carried out with link list

Disadvantage of Link List

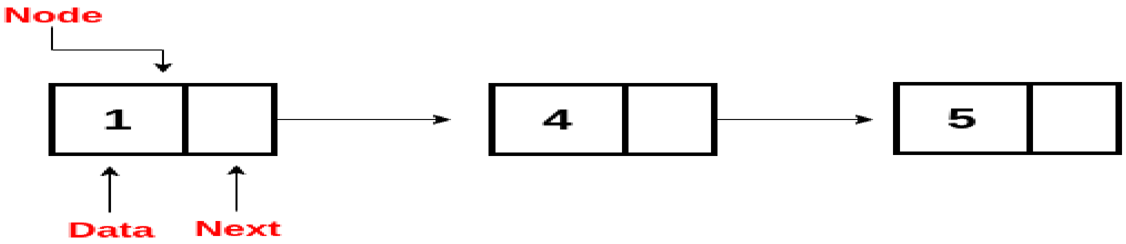
- More memory is required
- Nodes traversal is difficult in linked list.
- In linked list each node contains a pointer and it requires extra memory for itself.

Operation on Linked List

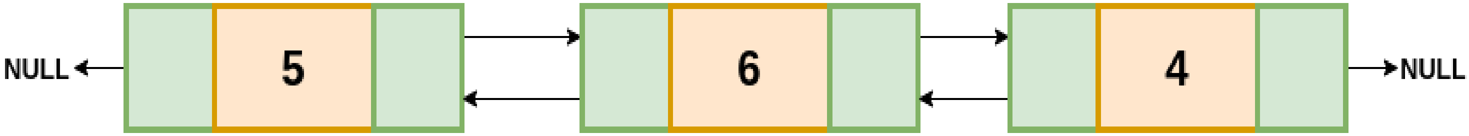
- **Insertion** − Adds an element at the beginning of the list.
- **Deletion** − Deletes an element at the beginning of the list.
- **Display** − Displays the complete list.
- **Search** − Searches an element using the given key.
- **Delete** − Deletes an element using the given key.

Types of Linked List

1. Simple Linked List – Item navigation is forward only.



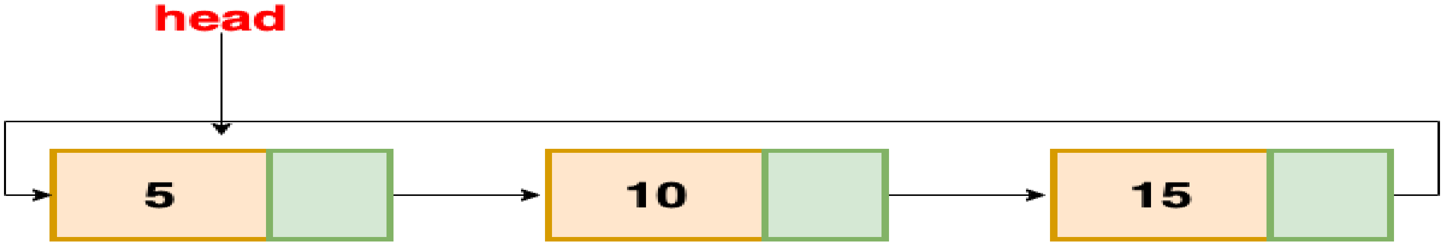
2. Doubly Linked List – Items can be navigated forward and backward.



Doubly Linked List

3. Circular Linked List

- A circular linked list is either a singly or doubly linked list in which there are no ***NULL*** values.



Circular Linked List

Representation of structure of linked List

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