

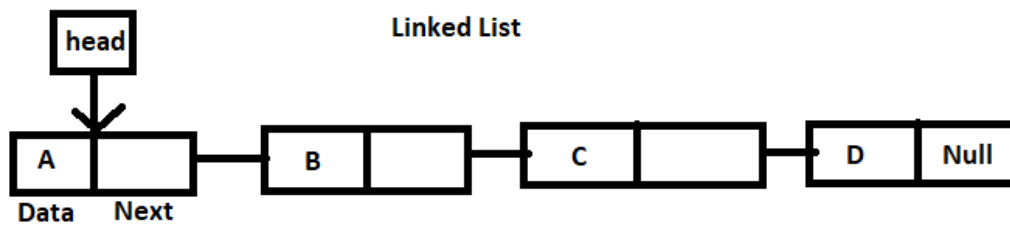
Session 1

1. Linked List

Linked List is collection of nodes where each node contains data and reference to next node

There are three types of Linked list 1.Singly,Doubly and Circular

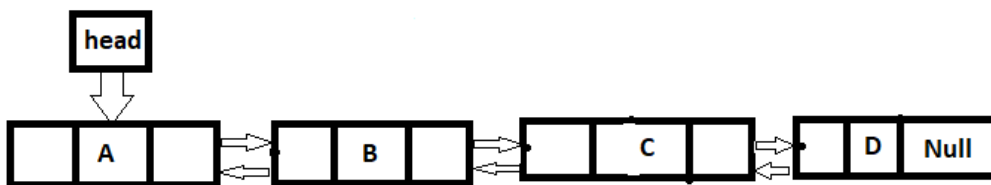
2.Singly Linked List



Advantages:

- it is very easier for the accessibility of a node in the forward direction.
- the insertion and deletion of a node are very easy.

3.Doubly Linked List

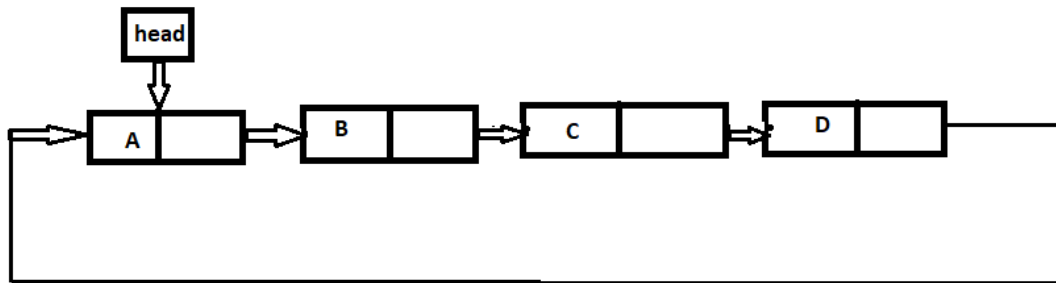


In doubly Linked list ,each node also have the address to previous node.

Advantages.

- Allows to iterate in both directions.
- We can delete a node easily as we have access to its previous node.
- Reversing is easy.

4.Circular Linked List

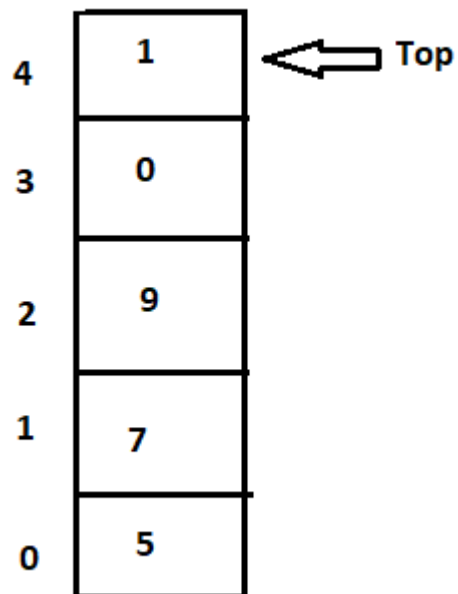


in the circular linked list the last node of the list point to the address of the head.

Advantages

- Easily we can go to head from the last node
- In a circular list, any node can be starting point means we can traverse each node from any point.

5.Stack



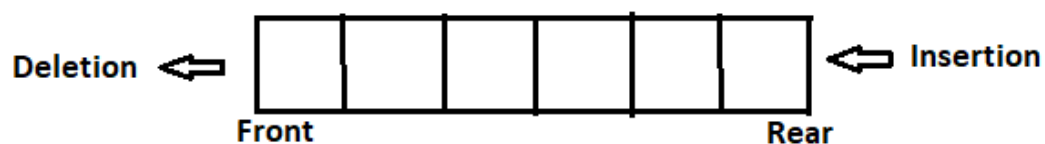
Stack follows LIFO order. Only top element is accessible.

Insertion and deletion takes from top.

Advantages:

- Stack helps you manage the data in a LIFO

6.Queue



Queue follow FIFO order .Insertion takes place at rear end and deletion take place from front end.

Advantages

-maintain data in FIFO manner

Session 2

1.Linear Search

```
1
2 public class LinearSearch {
3 public static void main(String[] args) {
4     int arr[] = {1,2,3,4,5,6,7,8,9};
5     int key=7;
6     for(int i=0;i<arr.length;i++) {
7         if(arr[i]==key) {
8             System.out.println("Element found at index "+i);
9             break;
10        }
11    }
12 }
13 }
14 }
```

Problems @ Javadoc Declaration Progress Console

<terminated> LinearSearch [Java Application] C:\Program Files\Java\jdk1.8.0_261\bin\javaw.exe (

Element found at index 6

2.Binary Search

```

1 class BinarySearch {
2     public static int binarySearch(int arr[],int key) {
3         int low=0,high=arr.length-1,mid;
4         while(low<=high) {
5             mid=low+(high-1)/2;
6             if(key==arr[mid])
7                 return mid;
8             if(key>arr[mid])
9                 low=mid+1;
10            else
11                high=mid-1;
12        }
13        return -1;
14    }
15
16
17    public static void main(String[] args) {
18        int arr[] = {1,2,3,4,5,6,7,8,9};
19        int key=4;
20        int j=BinarySearch.binarySearch(arr, key);
21        System.out.println("Element found at index "+j);
22    }
23 }

```

Problems @ Javadoc Declaration Progress Console

<terminated> BinarySearch [Java Application] C:\Program Files\Java\jdk1.8.0_261\bin\javaw.exe (10 Feb, 20
Element found at index 3