Linked List

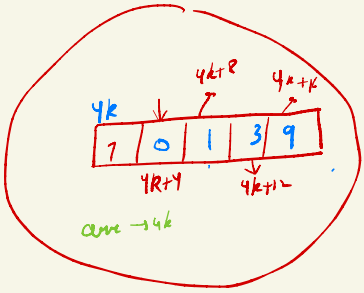
arr(0) → 4k arr → 4k

4k → 4k+4 →

int[] arr = new int[5];

arr → 4k 5*4

arr(0) 4
arr(1) → 4k+4



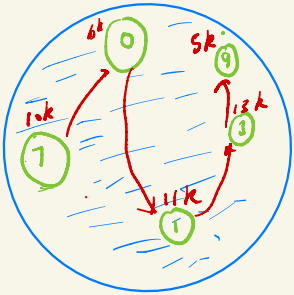
data
address of next Node object

20 bytes of autonomous space

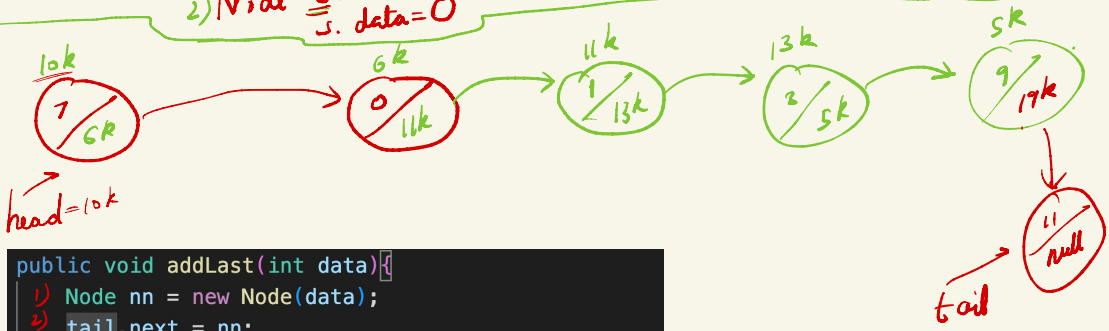
Student s

class Node {
→ int data;
→ Node next;
}

{ 1) Node f = new Node();
f.data = 7
2) Node s = new Node()
s.data = 0



s = 6k s = 6k




```java
public void addLast(int data){
    Node nn = new Node(data);
    tail.next = nn;

    tail = nn;
}
```

```java
public static void main(String[] args) {
    Node f = new Node(d: 7);
    Node s = new Node(d: 0);

    Node t = new Node(d: 1);

    // connecting first and second
    f.next = s;
    s.next = t;

    System.out.println(f.next.data); //0
}
```

head = 5k

head → 5k | 7 / 10k → 10k | 0 / 8k → 8k | 1 / 20k

break till 10:05

head = f
tail = t

```java
public void addLast(int data){
    Node nn = new Node(data);
    tail.next = nn;

    tail = nn;
}
```

⇒ 9

→ create new object

tail = 20k | 9 / null

⇒ 0 size

⇒ 1 size

⇒ x size list

```java
public class Main {
    static Node head;
    static Node tail;

    public static void addLast(int data){
        Node nn = new Node(data);

        if(head == null){ // in case of 0 size ll
            head = nn; // this is my first node of list
            tail = nn; // and also my last node of list
        } else {
            tail.next = nn; // attaching

            tail = nn; // shifting tail to last
        }
    }

    Run | Debug
    public static void main(String[] args) {
        head =null;
        tail = null;

        for(int i=1; i<=3; i++){
            addLast(i);
        }

        System.out.println(head.data);
        System.out.println(tail.data);
    }
}
```

0 size

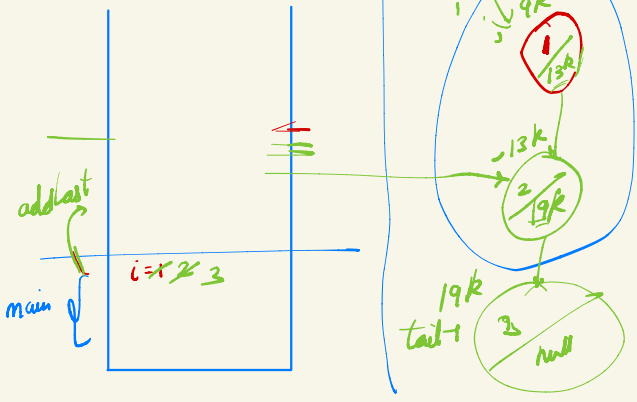
head = null

head = 9k

tail = 19k

i = 1

addlast (1)



Ques Add a node at the start of LL.

1) addFirst(1);
2) addFirst(2);
3) addFirst(3);

head = 5k
tail = 9k

```java
public static void addFirst(int data){
    Node nn = new Node(data);

    if(head == null){
        head = nn;
        tail =nn;
    } else {
        nn.next = head;
        head = nn;
    }
}
```