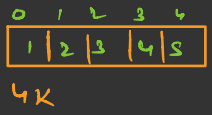
Linked list

```
int arr[] = new int[5]
```

20 bytes



Heap

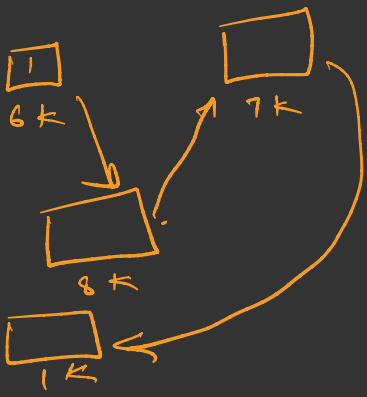
LL was introduced

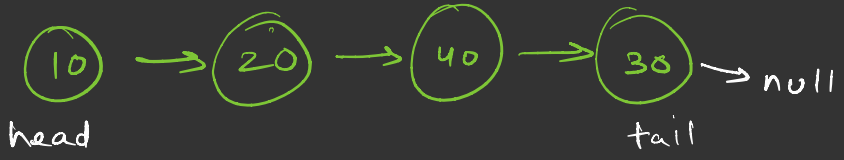
Node

1. data
2. reference to next node

Node

int data
Node next





```
Class LL {
    Node head
    Node tail
    int size
}
```

add

addFirst(int val)

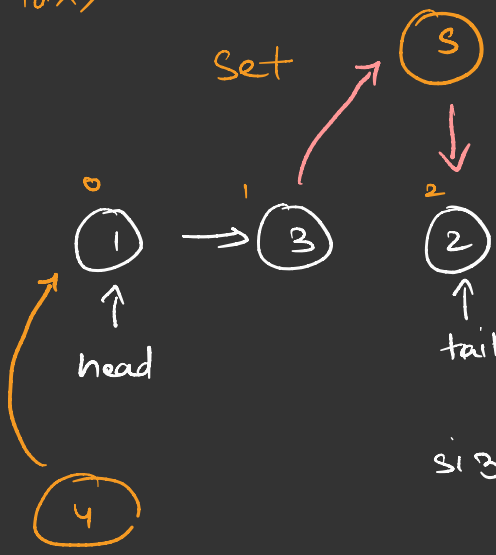
addLast(int val)

add(int val, int idx)

get

Remove

addFirst(4)



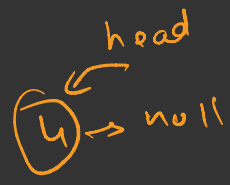
1. Create new node
2. point next to head
3. point head to new node
4. increase the size

size = 3

```
Node n = new Node(4);
n.next = head;
head = n;
```

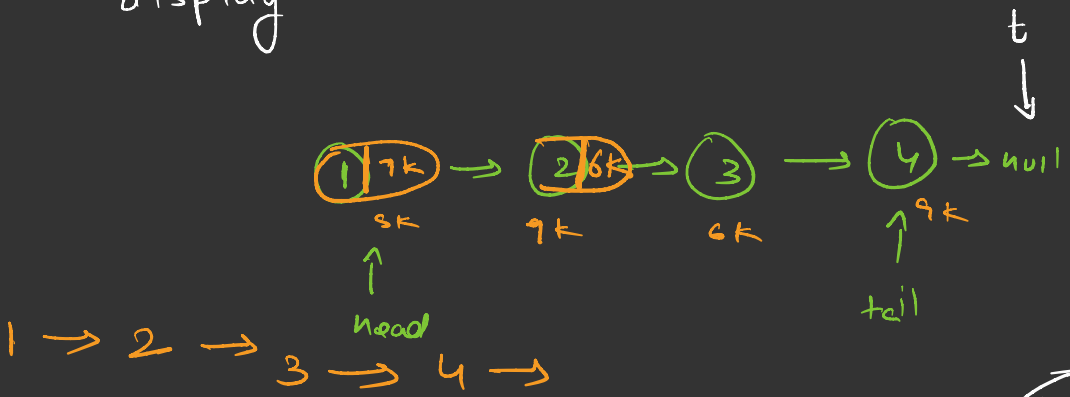
size == 0

head = null
tail = null
size = 0 + 1

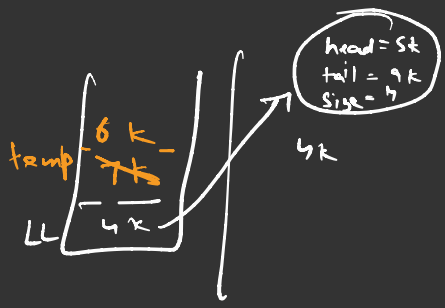


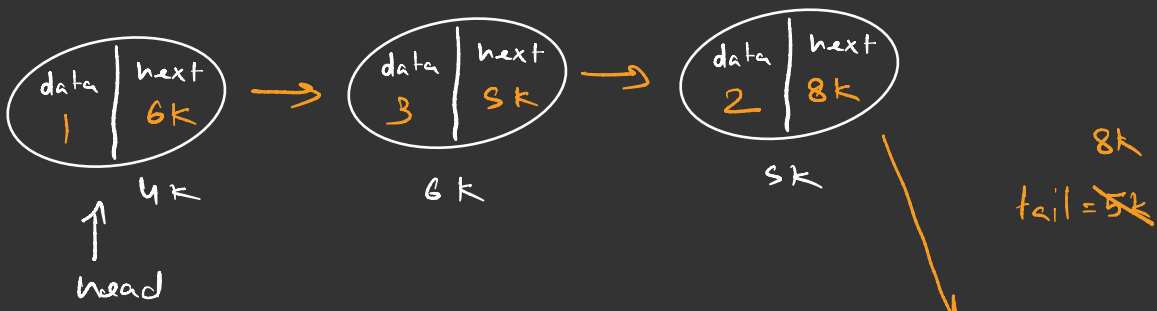
if (tail == null)
tail = n

display

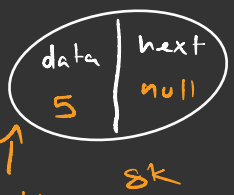



```java
public void display() {
    Node temp = head;
    while(temp != null) {
        System.out.print(temp.data + " ->
        temp = temp.next;
    }
    System.out.println("null");
}
```





addLast(3)



1) Create new node

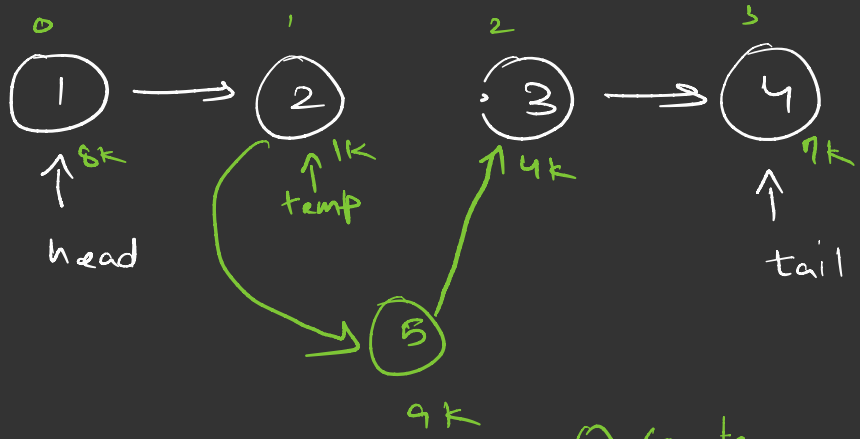
2) tail.next should point to new node (n)

3) tail should point to n

4) Increase size

add(int val, int idx)

add(5, 2)



temp = get(idx - 1)

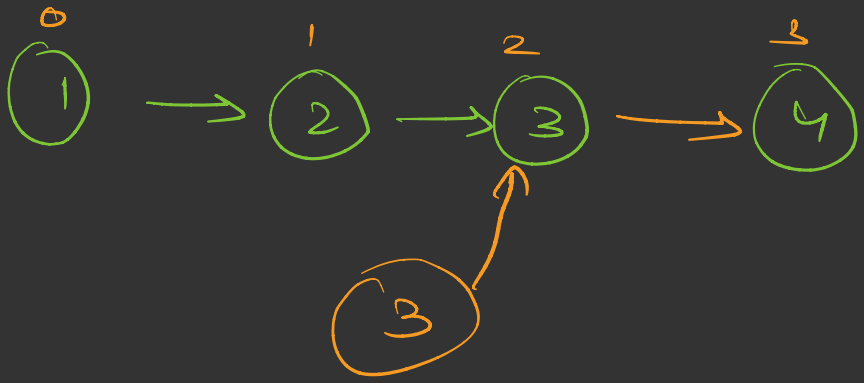
① Create

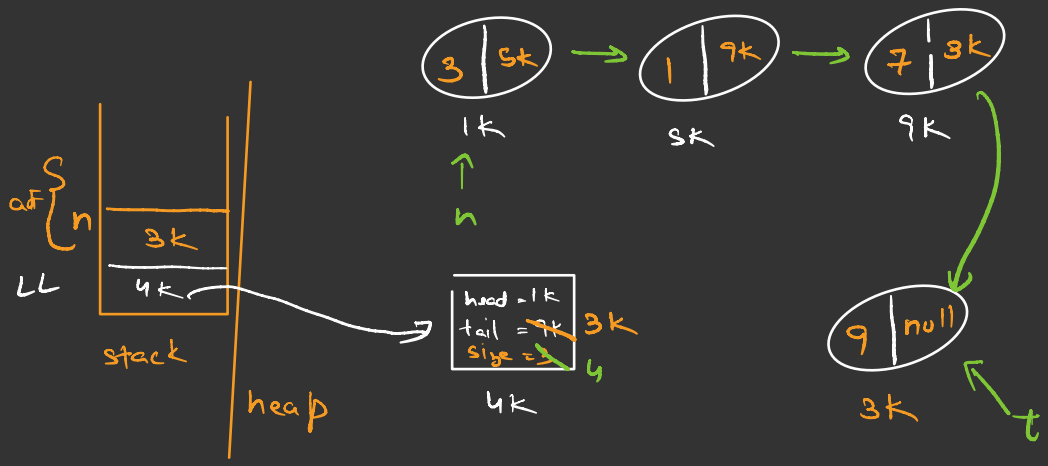
② get(idx - 1)

③ n.next = temp.next;

④ temp.next = n

⑤ size ++



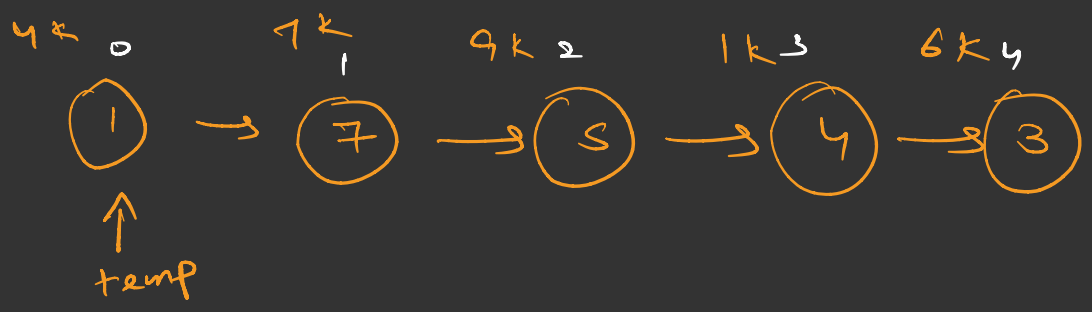



```java
public void addLast(int data) {
    if(this.size == 0) {
        this.addFirst(data);
        return;
    }
    Node n = new Node(data);
    this.tail.next = n;
    this.tail = n;
    this.size++;
}
```

---

get(idx)

getNode(idx)

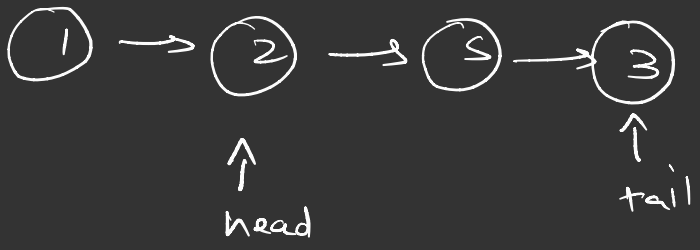


count = 0

idx = 3

temp = head = 4k

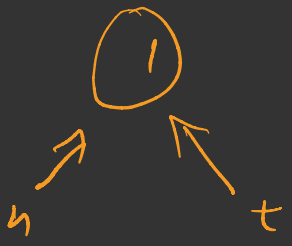
Remove First



1. Move head
2. reduce size

size = 0
return

size = 1



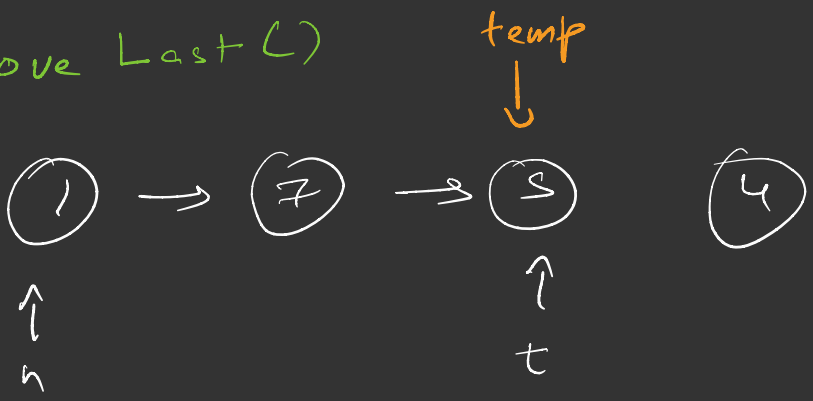
n

null

t = null

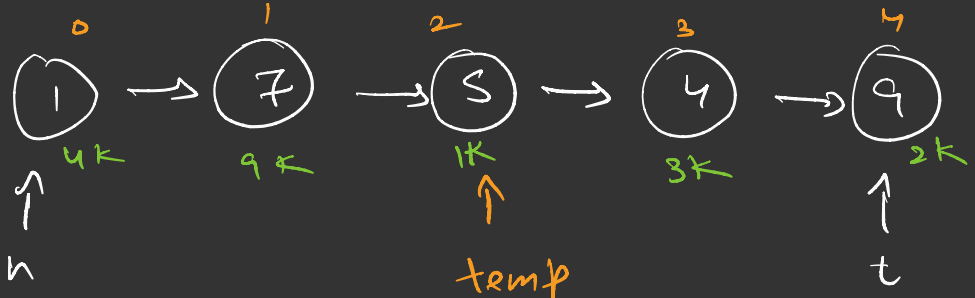
s--;

Remove Last()



1. temp = getNode(size - 2)
2. temp.next = null
3. tail = temp;
4. size--

Remove(int idx)



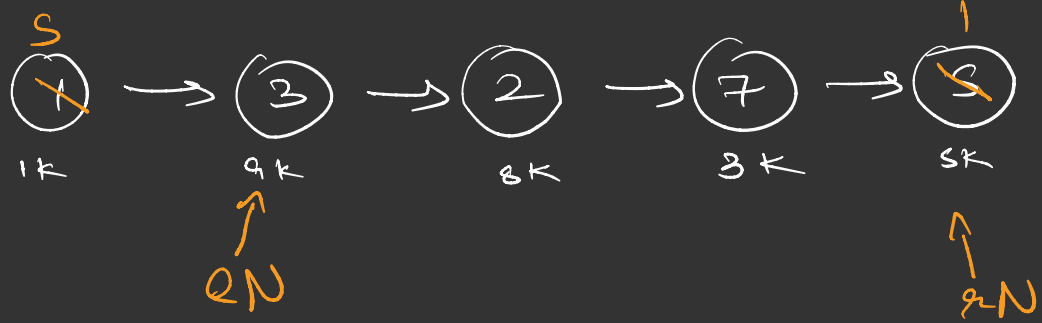
remove(3)

① getNode(idx - 1).

temp.next = temp.next.next

Reverse Data Iteratively

ReverseDI



r = 1

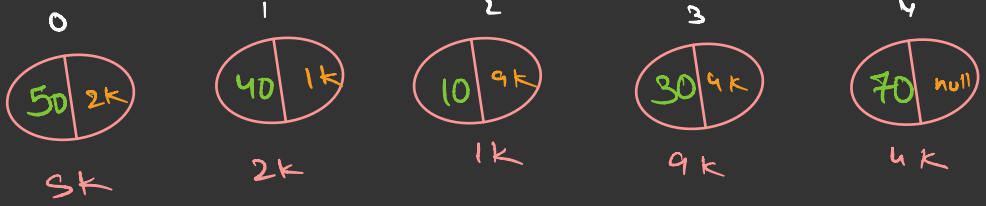
r = 3

Reverse

Data Pointer

```
Node get(Node head, int idx) {
  Node t = head;
  int c = 0;
  while( c < idx ) {
    t = t.next;
    c++;
  }
  return t;
}
```

get	c	3
	t	9k
	head	5k
	idx	3
main	head	5k



Ques Delete node without head pointer

D.data = D.next.data ;

D.next = D.next.next ;