Linked Lists

Agenda:

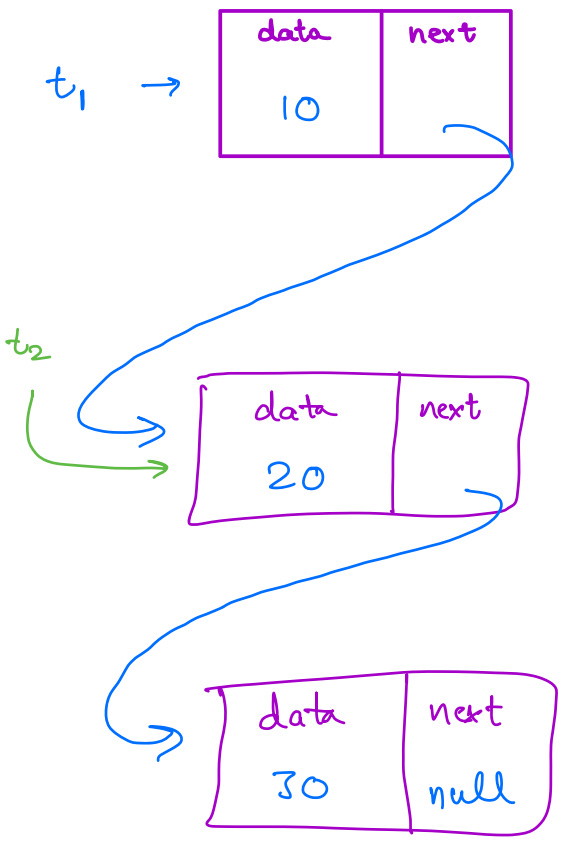
- Object Reference as Data Member
- Why Linked Lists ?
- Size
- Insertion
- Deletion

Classes & Objects

Object Reference as Data Member

```
class Node {
    int data
    Node next

    constructor( x ) {
        data = x
        next = null / None
    }
}
```



t_1 = new Node(10)

print(t_1.data) → 10

print(t_1.next) → None

t_1.next = new Node(20)

t_2 = t_1.next

print(t_2.data) → 20

print(t_1.data) → 10

print(t_1.next.data) → 20

$t_2.next = new\ Node(30)$

print(t_2.data) → 20

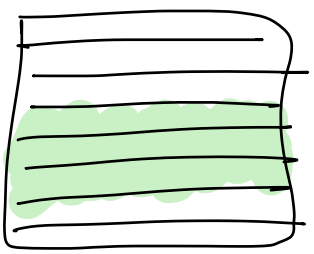
print(t_2.next.data) → 30

print(t_1.next.next.data) → 30

print(t_1.next.next.next) → null

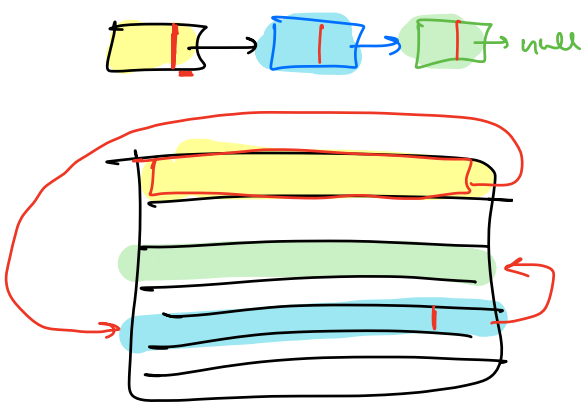
Array

arr[5]



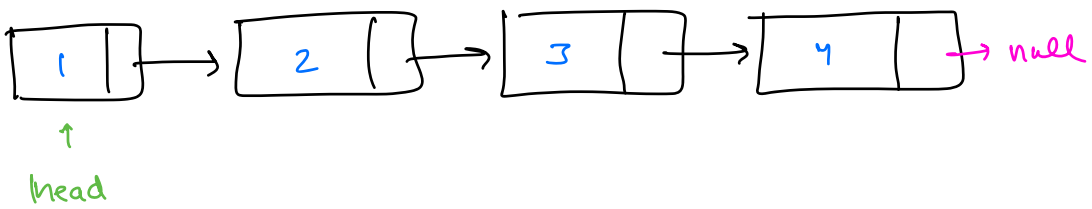
RAM

Linked List



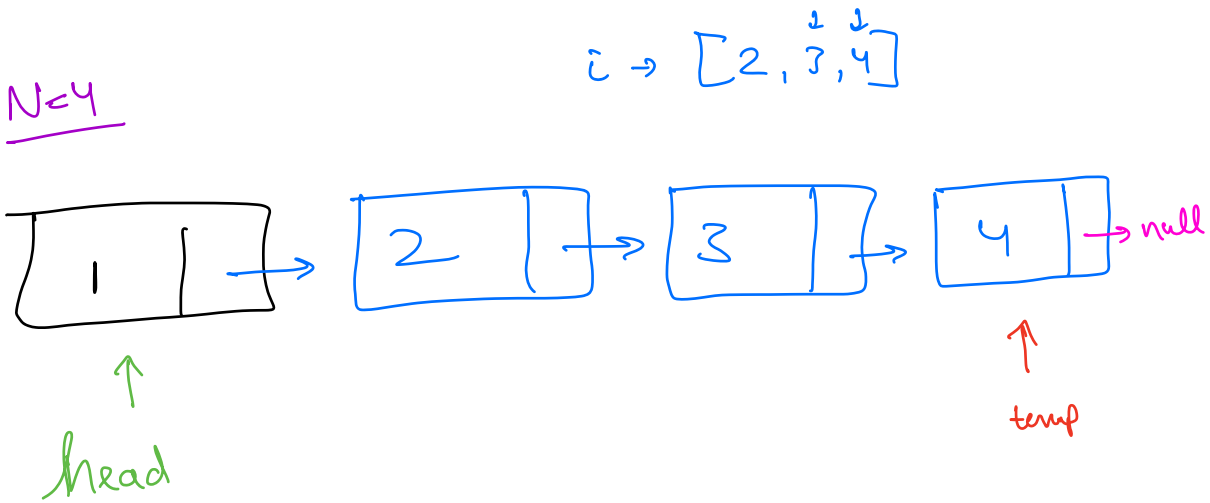
Q1 Given N > 0, create a linked list which contains data from 1 to N.

Example N = 4



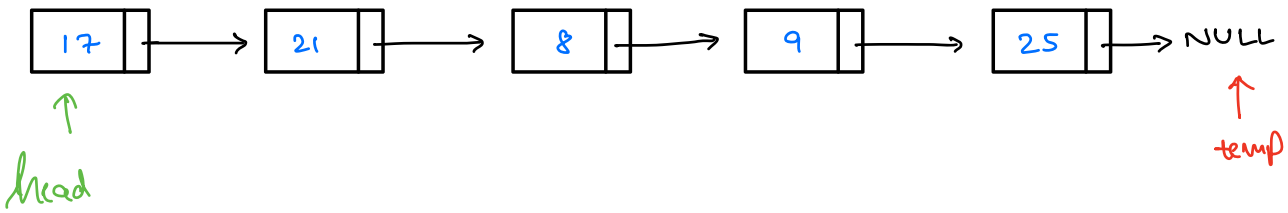
Nodes having data [2 to N]

```
Node createLL (int n) {
    Node head = new Node (1)
    Node temp = head
    for(i=2 ; i<=n; i++) {
        temp.next = new Node (i)
        temp = temp.next
    }
    return head
}
```


N=4
i → [2, 3, 4]
1
2
3
4
null
head
temp

Q2 Given head Node of a linked list, return its size.

Example



⇒ 5

c = 0 1 2 3 4 5

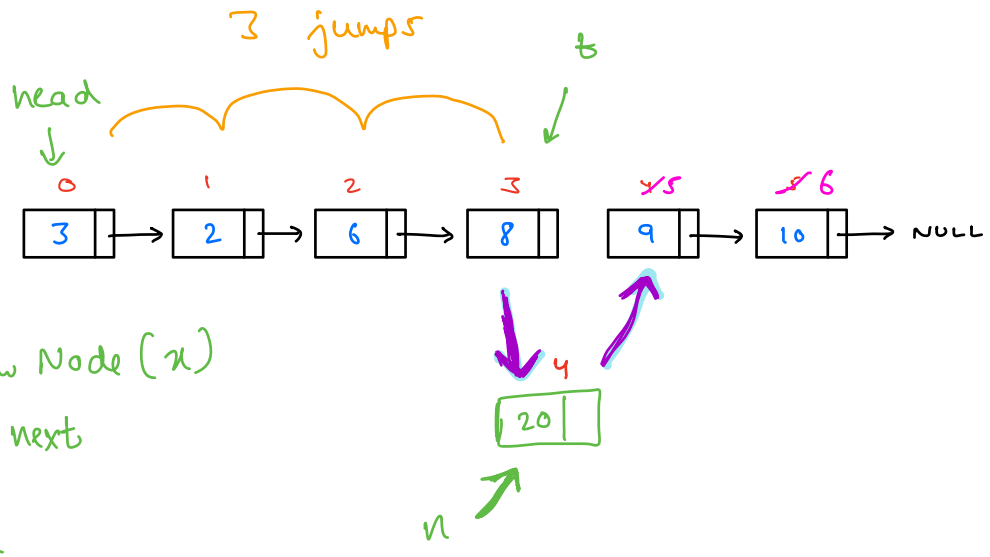
```
int size (Node head) {
    Node temp = head
    int c = 0
    while( temp != NULL) {
        temp = temp.next
        c = c+1
    }
    return c
}
```

Break till 10:13 PM

Q3 Given a head node of a LinkedList, insert a new node at kth index (0 based indexing)

Example 1

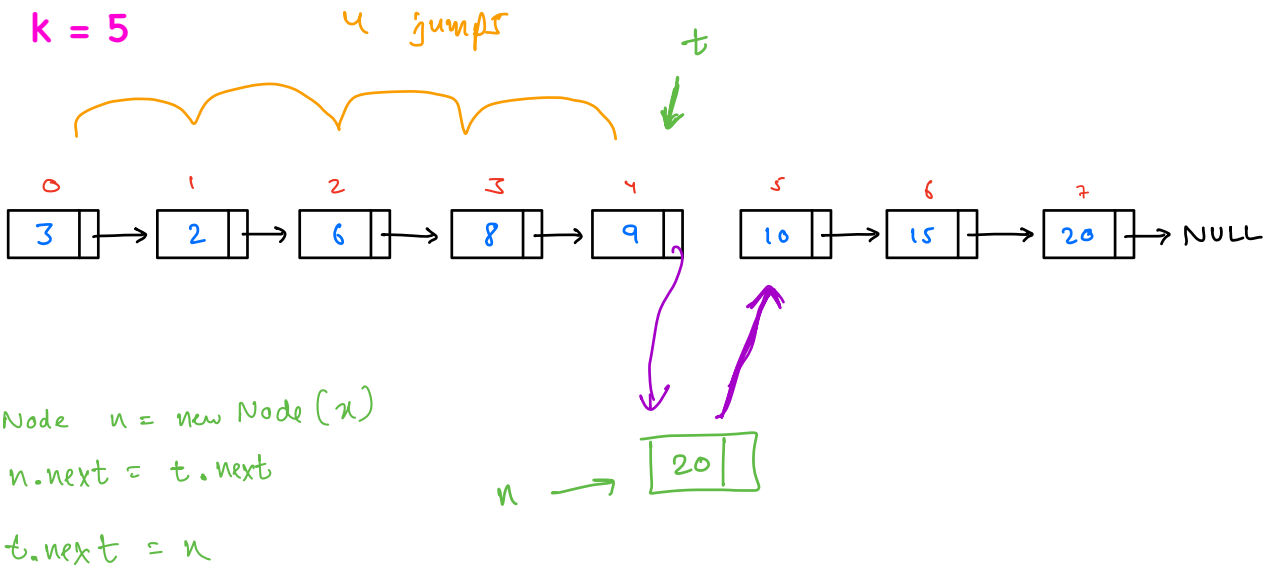
x = 20
k = 4




```
Node n = new Node(x)
n.next = t.next
t.next = n
```

Example 2

x = 20
k = 5

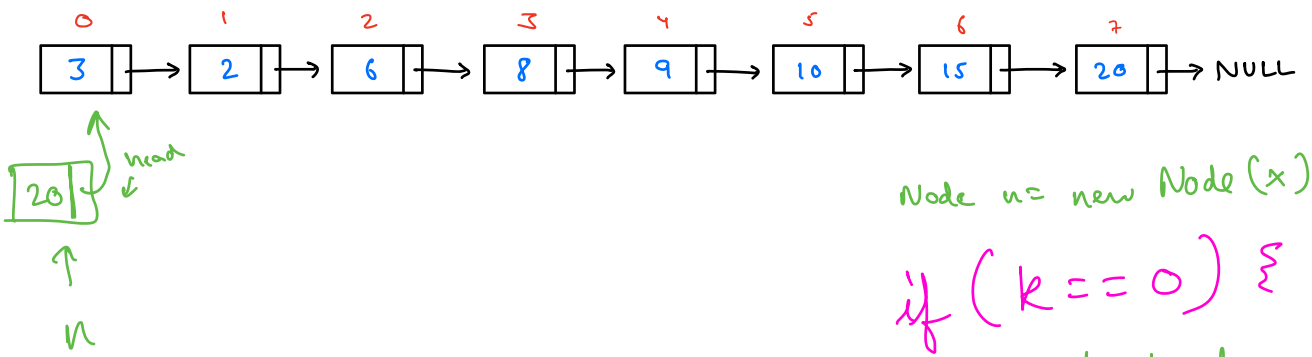



```
Node n = new Node(x)
n.next = t.next
t.next = n
```

Example 3

x = 20

k = 0




```
Node n= new Node(x)
if (k==0) {
    n.next= head
    head = n
}
```

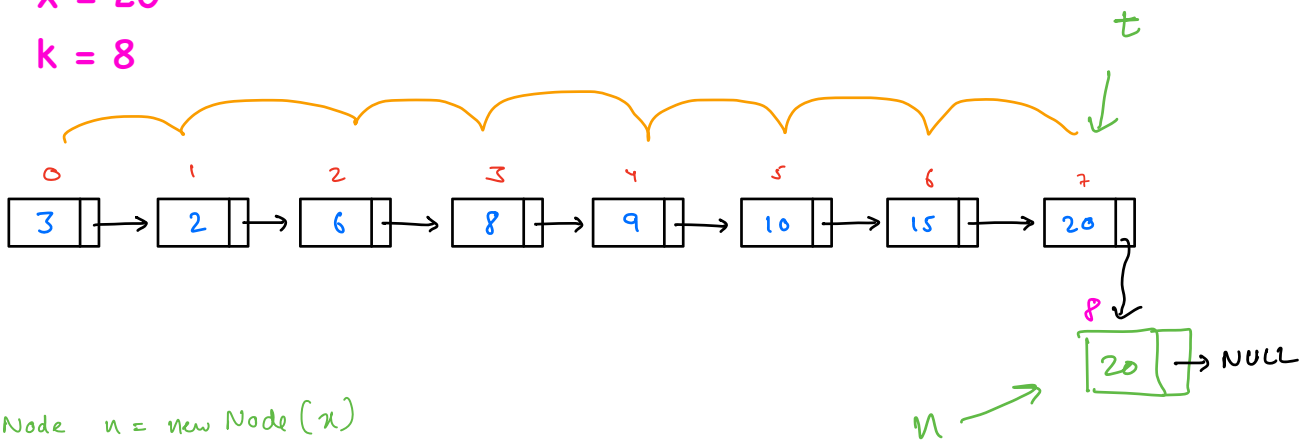
Insertion at Head

Example 4

7 jumps

x = 20

k = 8



Node n = new Node(x)

n.next = t.next

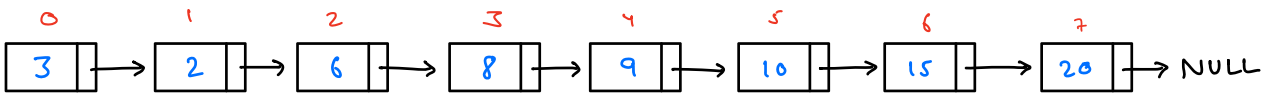
t.next = n

Insertion at tail

Example 5

x = 20

k = 9



Q2

if (k > size(head) or k < 0)

Not possible

return head

← Return LL as it is.

k = -10

```
Node insert ( Node head, int pos, int data) {
if (pos < 0  or  pos > size(head)) {
        return head
}
Node n= new Node (x)
if (pos == 0) {
        n.next= head
        head = n
        return head
}
Node t = head
for (i=0; i < pos -1 ; i++)        <- (k-1) jumps
        t = t.next

n.next = t.next
t.next = n
return head
}
```

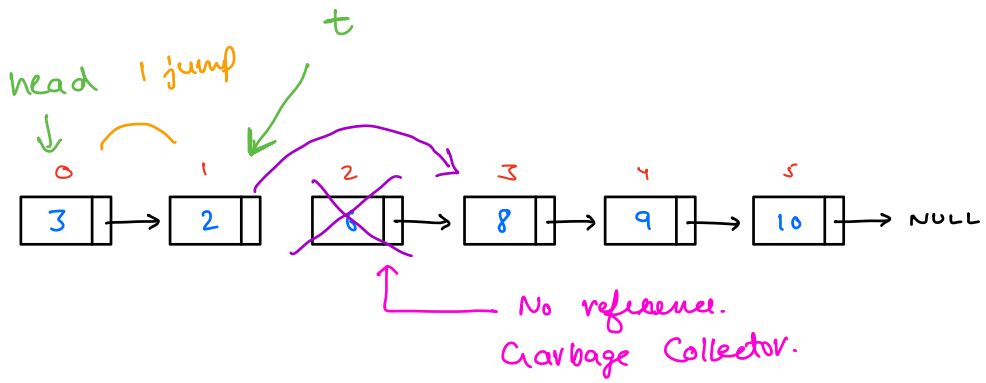
TC: O(N)

SC: O(1)

Q4 Given a head node of a LinkedList, delete the node at kth index.

Example 1

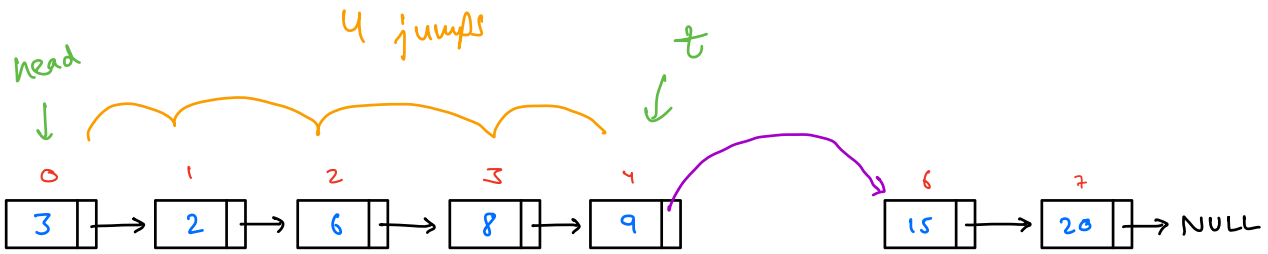
k = 2



t.next = t.next.next

Example 2

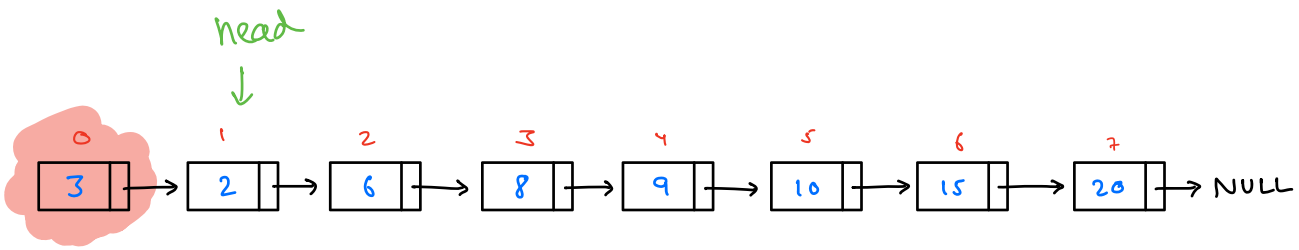
k = 5



t.next = t.next.next

Example 3

k = 0

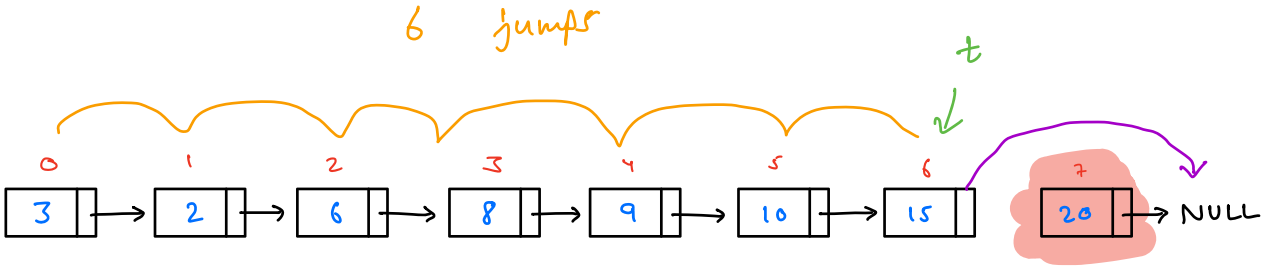


Deletion at head

```
if (k==0) {
    head= head.next
}
```

Example 4

k = 7



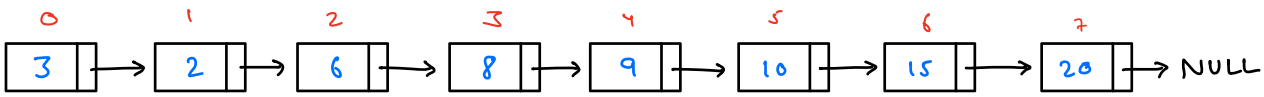
t.next = t.next.next

NULL

Deletion at tail

Example 5

k = 9



Not possible

```
if ( k < 0   or   k > size(head) )
    return head;
```

```
Node remove( Node head, int pos) {
if( pos < 0   or   pos > size(head) )
    return head

if( pos == 0 ) {
    head = head.next
    return head
}
```

```
Node t = head
for (i=0; i< k-1 ; i++)
        t = t.next

t.next = t.next.next

return head

}
```

(k-1) jumps

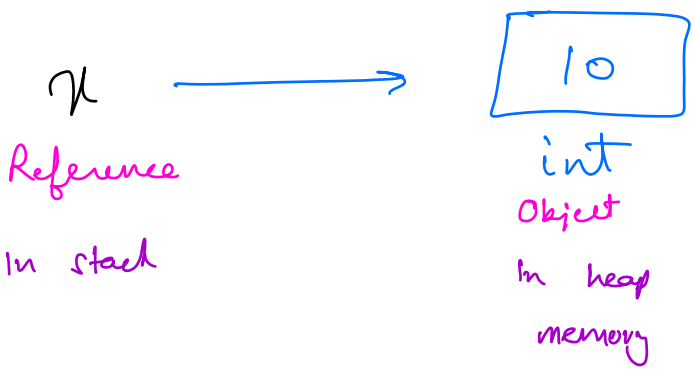
Slides =

https://slides.com/tarunluthra/linked-lists-basics-python

Doubts

Thank
You

$x = 10$



Usecases

→ Interviews

→ DBMS

30-35 problems

Good
Night

Thank
You

Monday