

Linked list : Basic Problems

Question

check if the value X is present in LL.

Searching

Array $\begin{cases} \rightarrow \text{unorganized} \\ \rightarrow \text{organized} \end{cases}$

$TC = O(N)$

$TC = O(\log N)$
binary search

Linear Search

Node temp = head

while (temp != NULL) {

if (temp.data == X)

return true

temp = temp.next

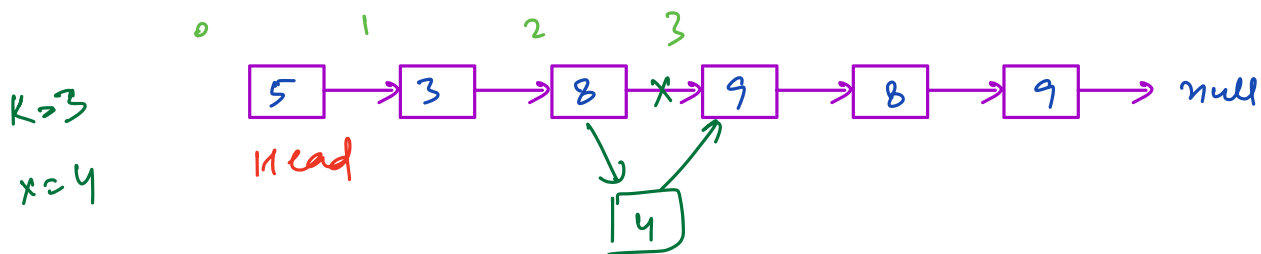
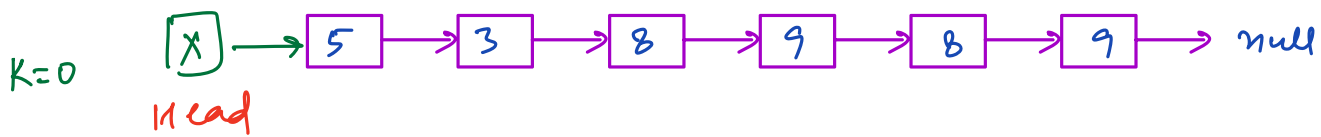
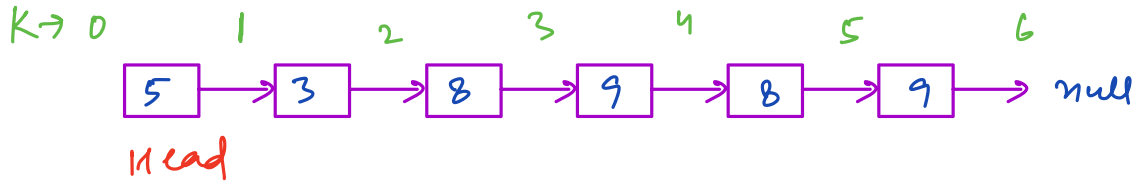
$TC = O(N)$

}

return false

Question

Insert a new node with data X at index K .
($0 \leq K \leq n$)



$x=10$ $1 \rightarrow 6 \rightarrow 7 \rightarrow 9 \rightarrow \text{null}$

$K=2$

\Downarrow

$1 \rightarrow 6 \rightarrow 10 \rightarrow 7 \rightarrow 9 \rightarrow \text{null}$

1. Head = null $\rightarrow K=0$ // empty list ✓
2. $K=0 \Rightarrow$ Head will update ✓

Code

Node newNode = new Node(x)

if (K == 0) {

newNode.next = head

head = newNode

}

else {

Node temp = head

for (i = 0 to K-2) { // K-1 times

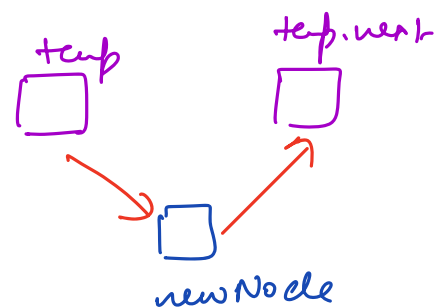
temp = temp.next

}

newNode.next = temp.next

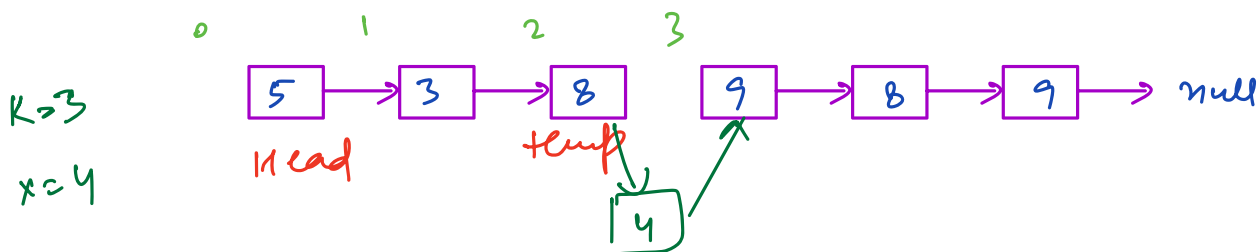
temp.next = newNode

}



TC = O(K)

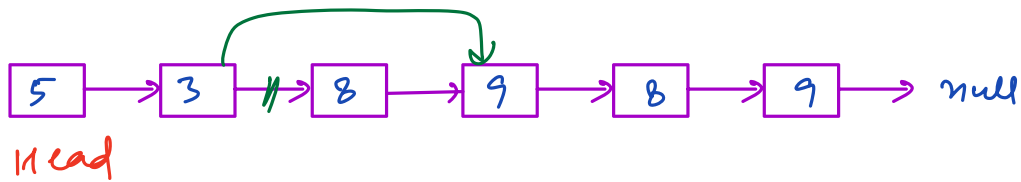
SC = O(1)



Question

Delete the first occurrence of value x from U .

If not present \Rightarrow no change



$x=8$

Casey:

1. Head = null ✓
2. Head.data = $x \Rightarrow$ delete head node ✓
3. No change // x not present ✓

$5 \rightarrow 4 \rightarrow 7 \rightarrow 1 \rightarrow \text{null}$

$x=1$

\Downarrow

$5 \rightarrow 4 \rightarrow 7 \rightarrow \text{null}$

Code

```
if (Head == null) return;
```

```
if (Head.data == x) {
```

```
    temp = Head
```

head = head.next

free(temp) → in java automatically done by GC
return but in other lang. it is needed

}

temp = head

while(temp.next != null) {

if(temp.next.data == x) {

p = temp.next

temp.next = temp.next.next

free(p)

break;

}

temp = temp.next

}

TC = O(N)

SC = O(1)

Ques remove defects

Remove all phones with defective model.

↓

Delete all occurrences of value X in LL.

code

```
if (Head == null) return;  
while (Head != null && Head.data == x) {  
    temp = Head  
    Head = Head.next  
    free(temp)  
}
```

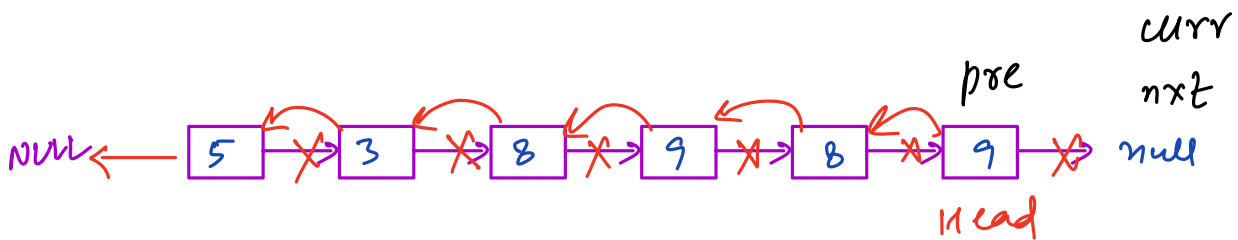
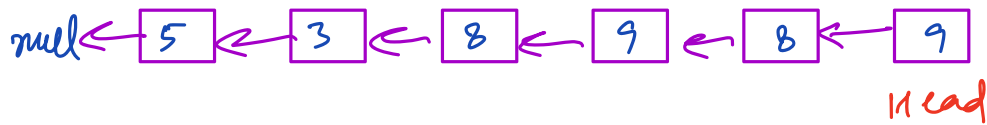
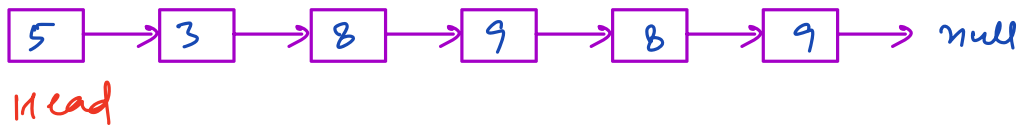
```
temp = Head  
while (temp != null && temp.next != null) {  
    if (temp.next.data == x) {  
        p = temp.next  
        temp.next = temp.next.next  
        free(p)  
    }  
    else {  
        temp = temp.next  
    }  
}
```

TC = O(N)

SL 200)

Question

Reverse the LL.



$curr.next = pre$

$pre = curr$

$curr = next$

$next = curr.next$

$5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$

\Downarrow

$5 \leftarrow 6 \leftarrow 7 \leftarrow 8 \leftarrow 9$

Code

pre = NULL

curr = Head

while (curr != NULL) {

 next = curr.next

 curr.next = pre

 pre = curr

 curr = next

}

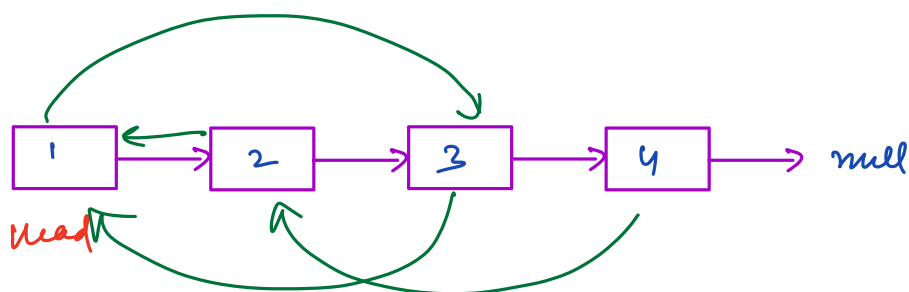
Head = pre

TC = O(N)

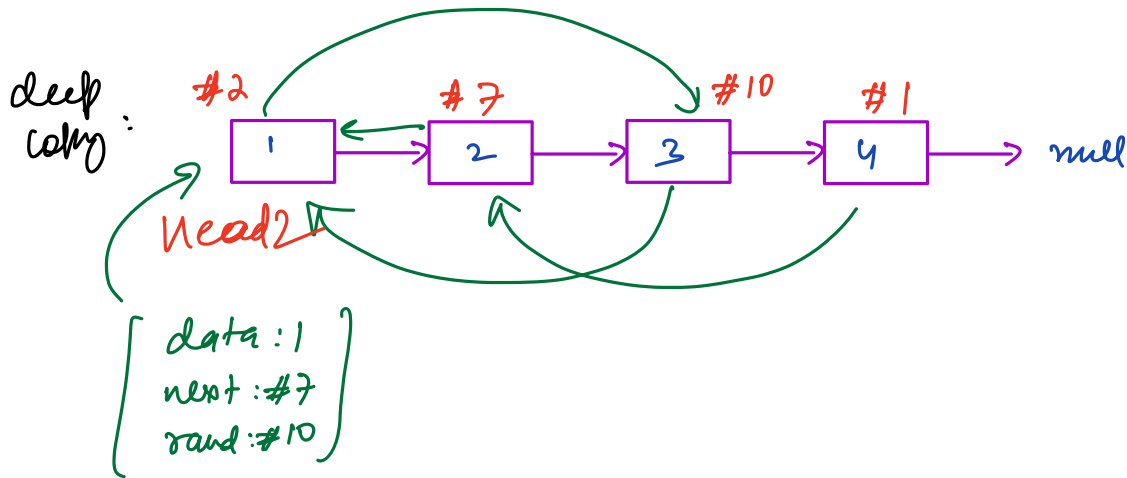
SC = O(1)

Question

Given a LL with next & rand pointer. Create a deep copy of this LL.



```
class Node {  
    int data;  
    Node next;  
    Node rand;  
}
```

Deep copy w/o rand pointer

Node head2 = new Node(head.data)

temp = head

temp2 = head2

while(temp.next != NULL) {

temp2.next = new Node(temp.next.data)

temp = temp.next

temp2 = temp2.next

}

(Handle corner cases)

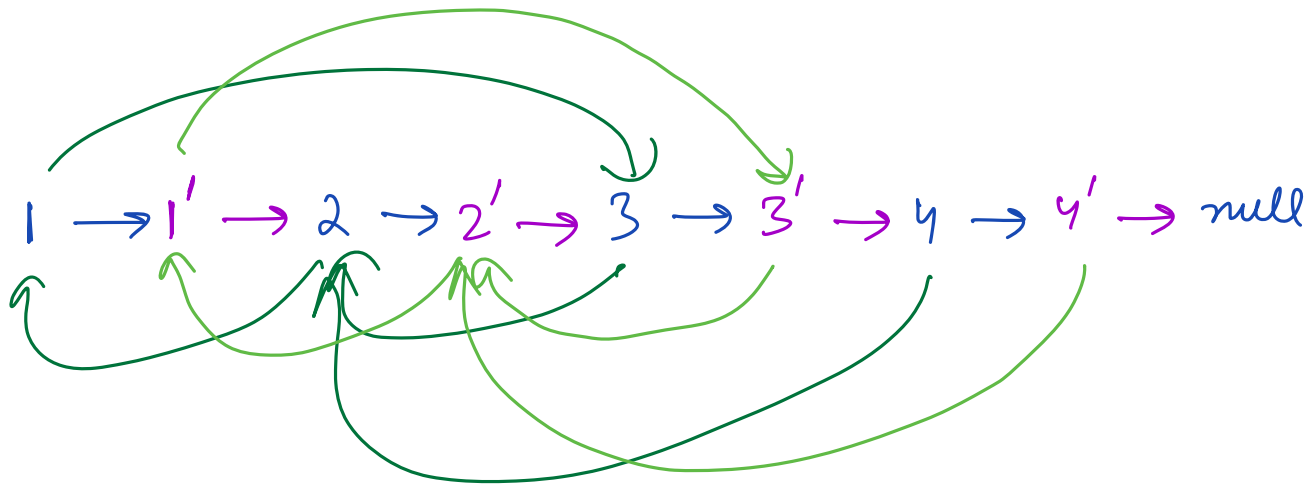
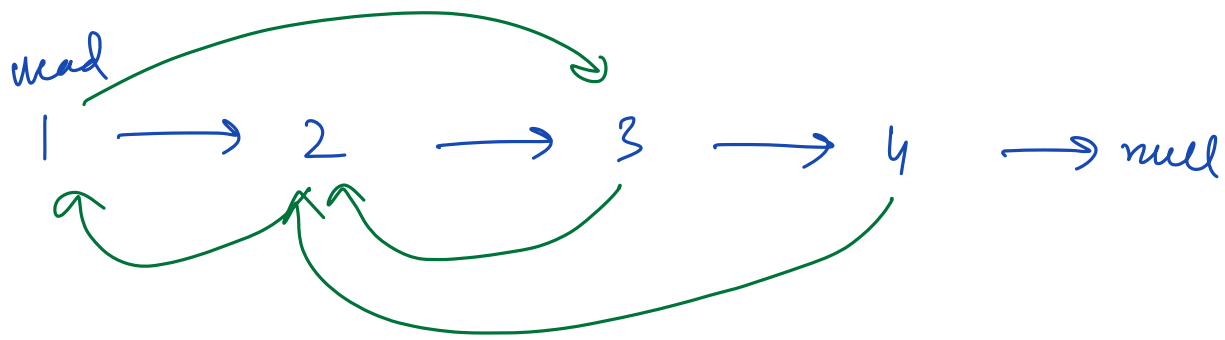
Ideal: HashMap < node, copy of node >

HashMap<Node, Node> hm

Node head2 = new Node(head.data)

temp = head

temp2 = head2



$t.\text{next}.\text{rand} = t.\text{rand}.\text{next}$

Detach logic

$u_2 = \text{head}.\text{next}$

$t_1 = \text{head}, \quad t_2 = u_2$

$\text{while} (t_1 \neq \text{null}) \{$

$t_1.\text{next} = t_1.\text{next}.\text{next}$

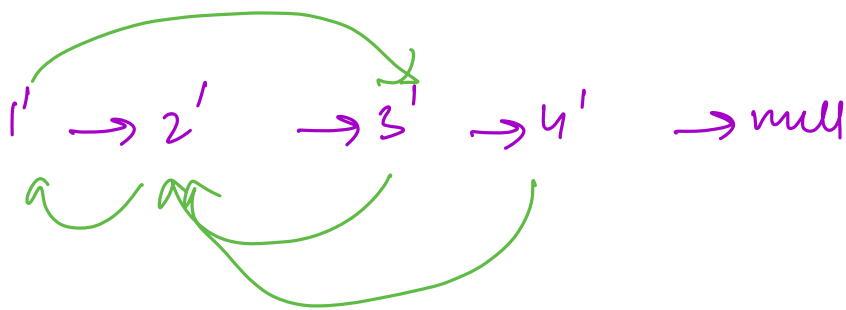
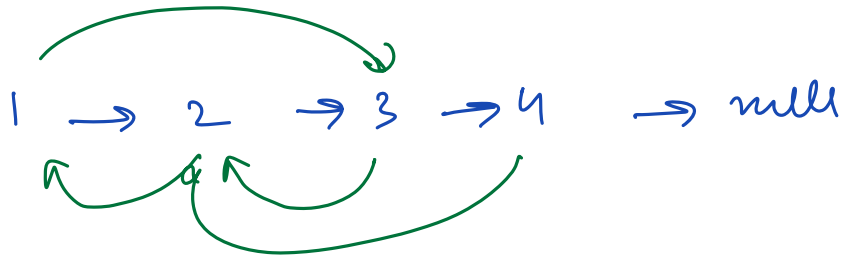
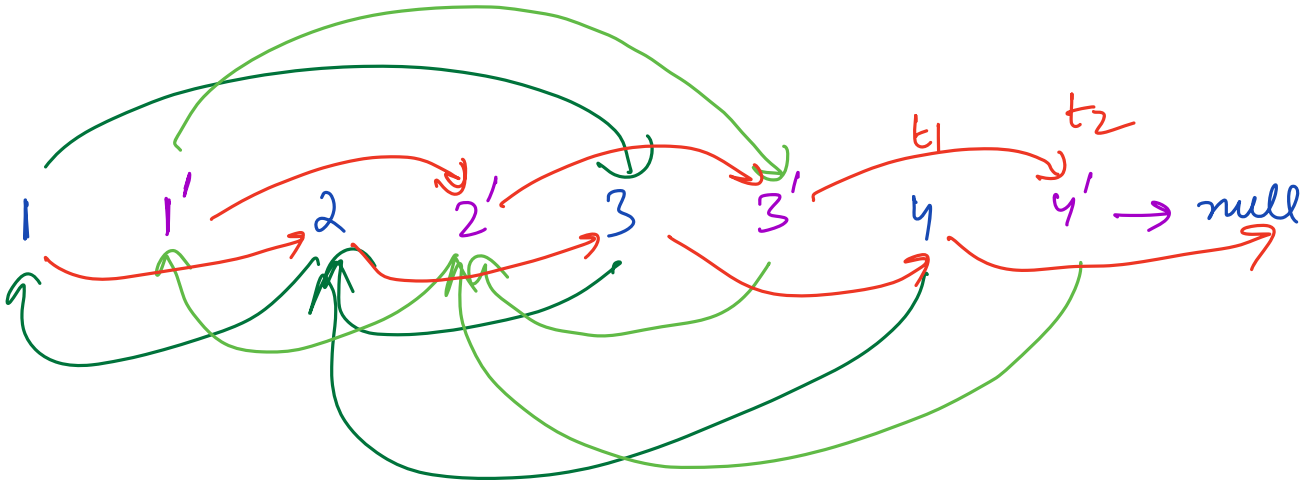
$\text{if} (t_2.\text{next} \neq \text{null}) \{$

$t_2.\text{next} = t_2.\text{next}.\text{next}$

3
 $t_1 = t_1 \cdot \text{next}$

$t_2 = t_2 \cdot \text{next}$

3



total TC = $O(N)$

SC = $O(1)$