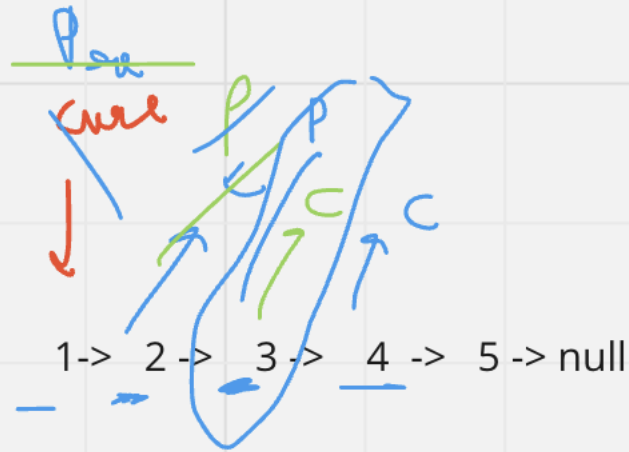


4

~~Null~~



Node curr = head;

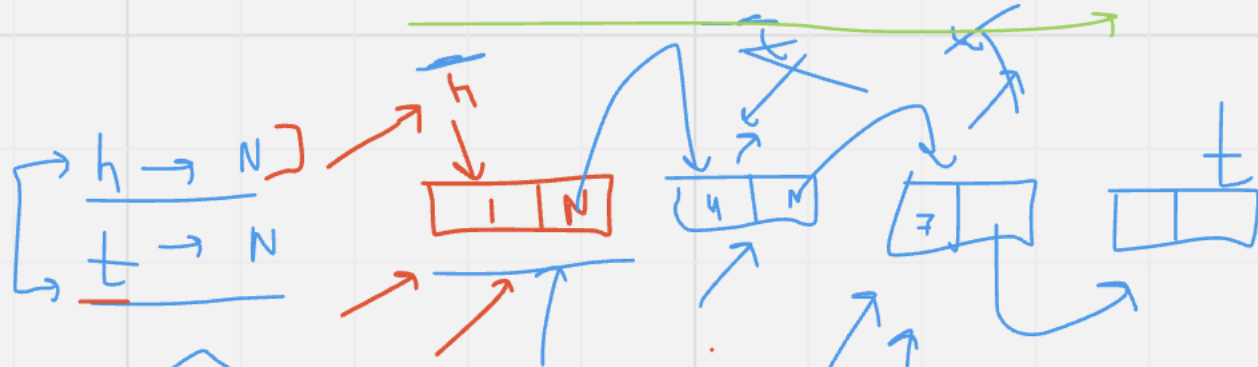
Node prev = null;

while(curr != null) {

if(curr.data == val) {  
prev = curr  
break

prev = curr;  
curr = curr->next;

}



✓ empty

if(head == null)

Node head = new Node(1);  
temp = head;

non - empty

Node newNode = Node(4);

temp.next = newNode;  
temp = temp.next;

public void addToTheLast(Node node)

{  
if (head == null)

{  
head = node;

tail = node;

}

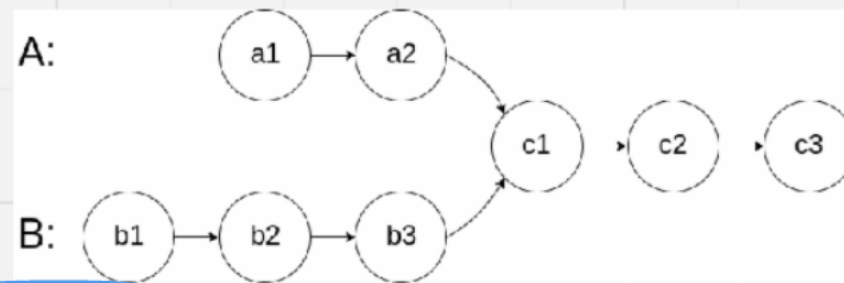
else

{  
tail.next = node;

tail = node;

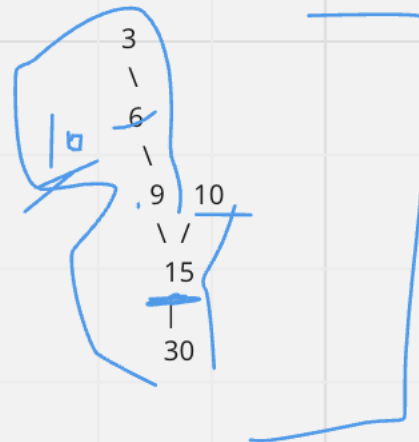
}

}



L1: 3 - 6 - 9 - 15 - 30

L2: 10 - 15 - 30



1 5 2 7 6

8 7 6

```
while(headA != null) {
```

```
    Node temp = headB;
```

```
    while(temp != null) {
```

```
        if(headA == temp) {  
            return headA;
```

```
        }else {  
            temp = temp.next;  
        }
```

```
    }
```

```
    headA = headA.next;
```

```
}
```

1 5 2 7 6

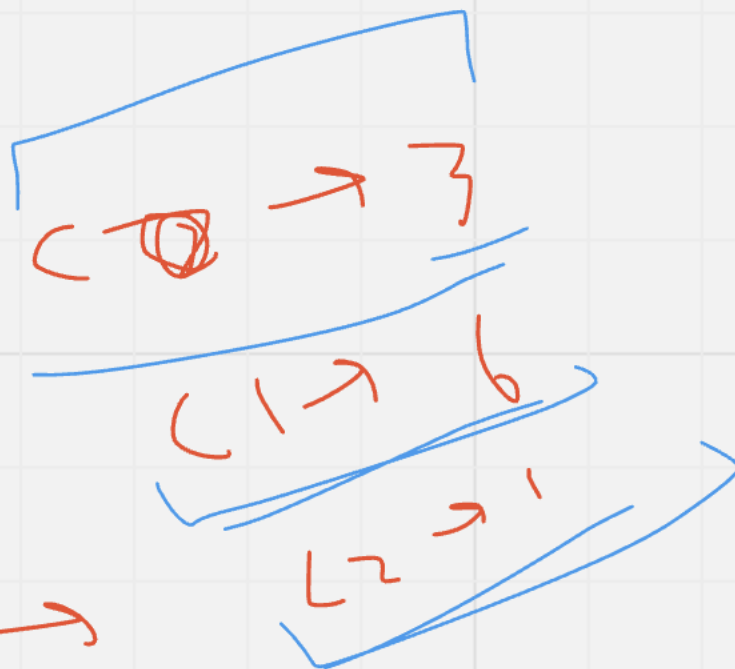
8 7 6

1 → 2 → 3 → 5 → N

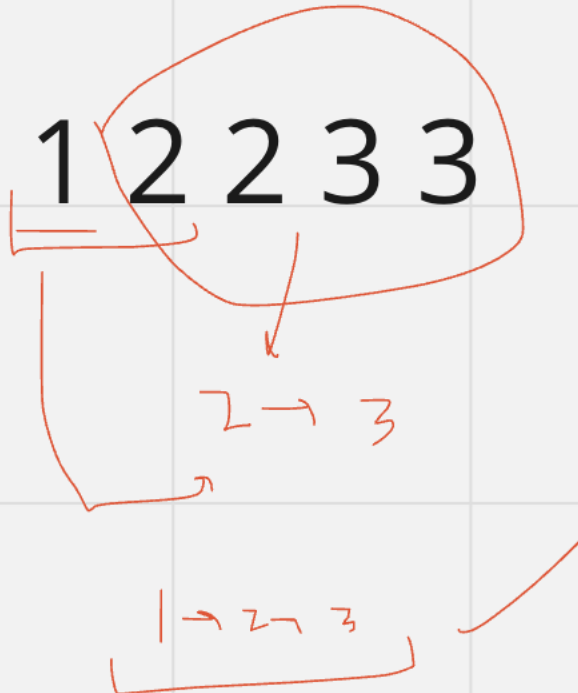
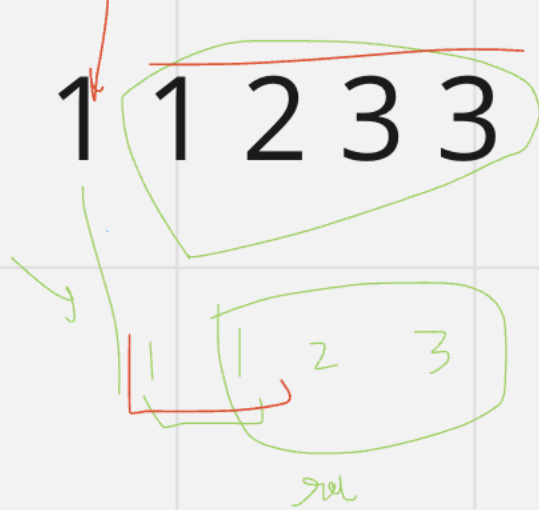
1 → 1 → 2 → 2 → 2 → 2 → 3 → 5 → null

1 → ~~1~~ → 2 → ~~2~~ → 3 → null

10  
0111201011  
0001111112

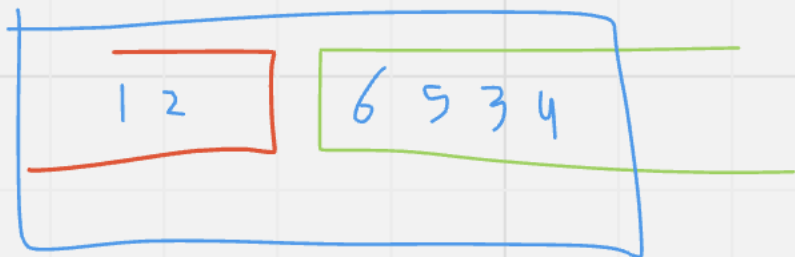


1 1 2 3 3

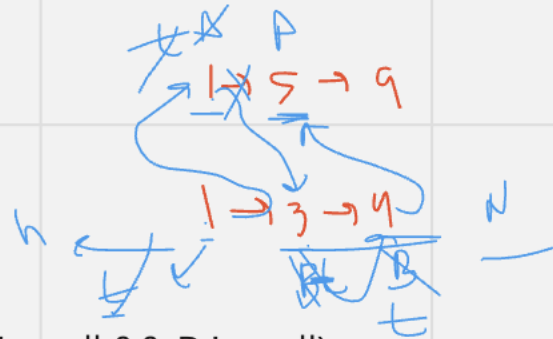


6  
1 6 2 5 3 4  
3 6 5 3 4

A handwritten diagram on grid paper. The top row contains the numbers 1, 6, 2, 5, 3, 4. Above the 2 is a 6. Below the 1 is a 3. Below the 6, 5, 3, 4 is another row of 6, 5, 3, 4. A red arrow points down to the 3 in the second row. Blue lines connect the 6 in the top row to the 6 in the second row, and the 5 in the top row to the 5 in the second row. A green arrow points left to the 3 in the top row.







```
while(A != null && B != null)
```

```
{
```

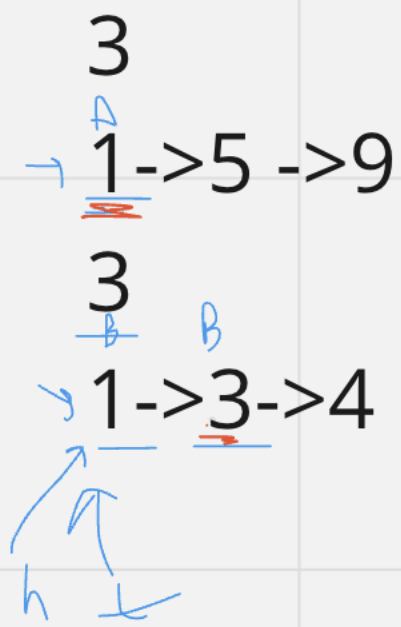
```
if(A.data < B.data) {
```

```
temp.next = A;
```

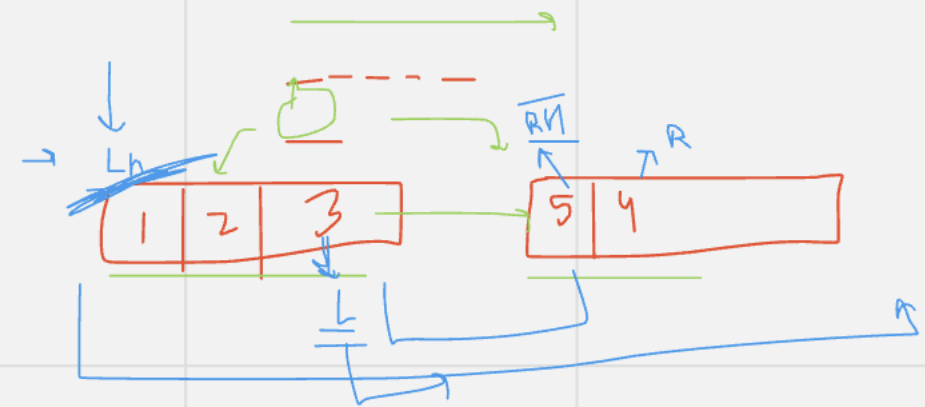
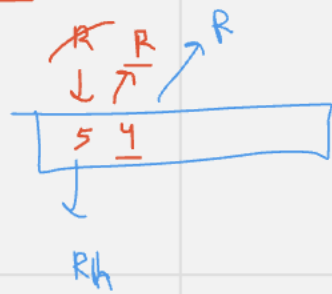
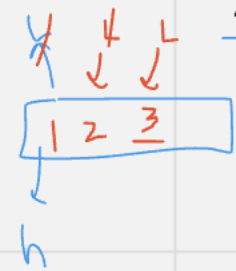
```
temp = temp.next;
```

```
}
```

```
}
```



5  
1 5 2 4 3  
4



```
h -> N
t -> N
if ( A.data < B.data)
{
    head = A;
    temp = head;
}
else {
    head = B;
    temp = head;
}
```