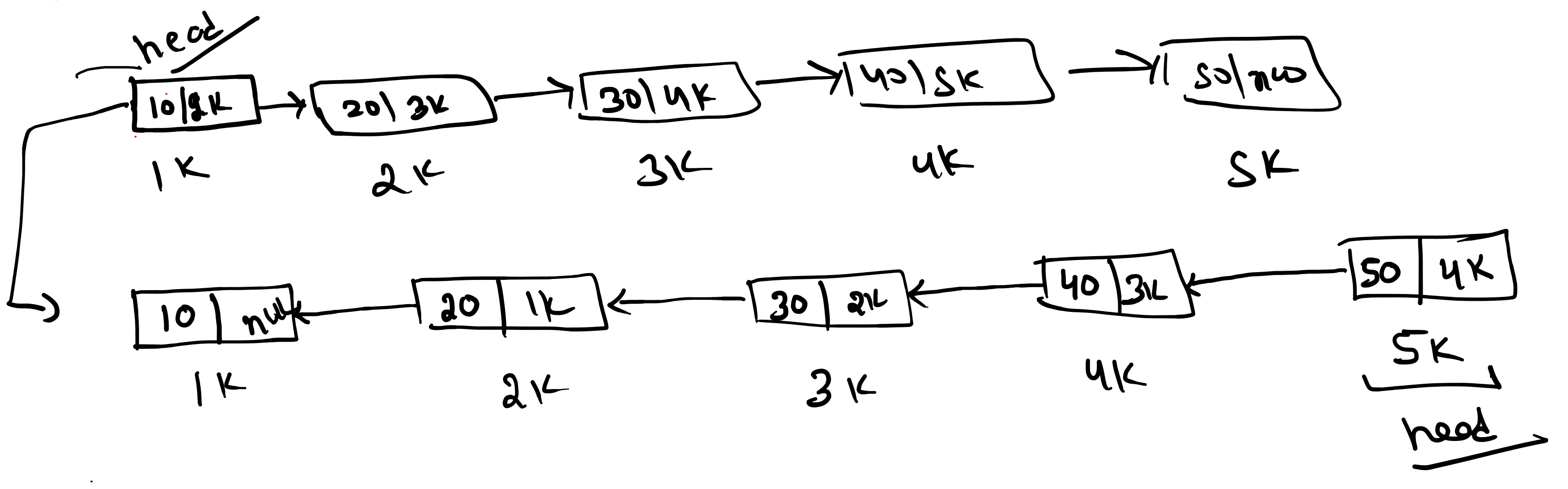
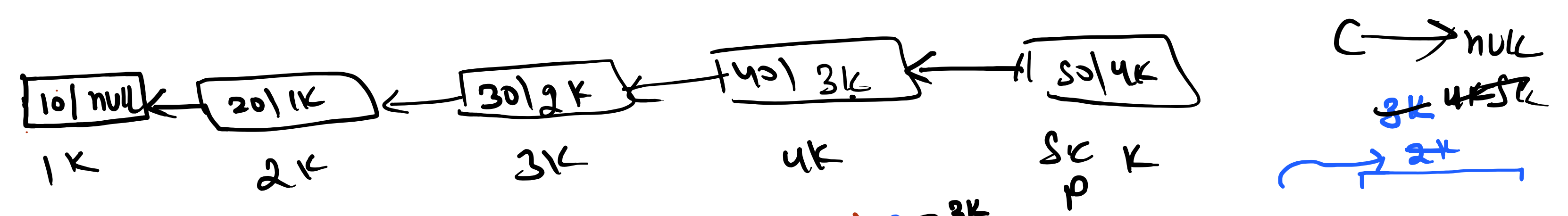
Lec-31

18/01/25 12:57 PM



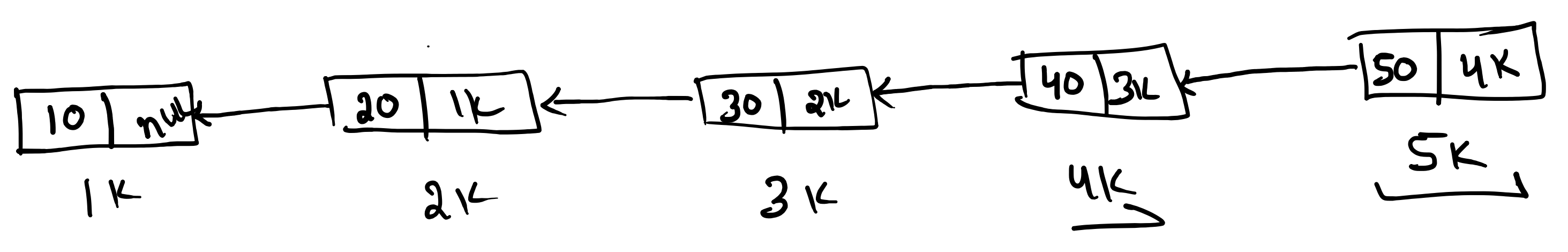


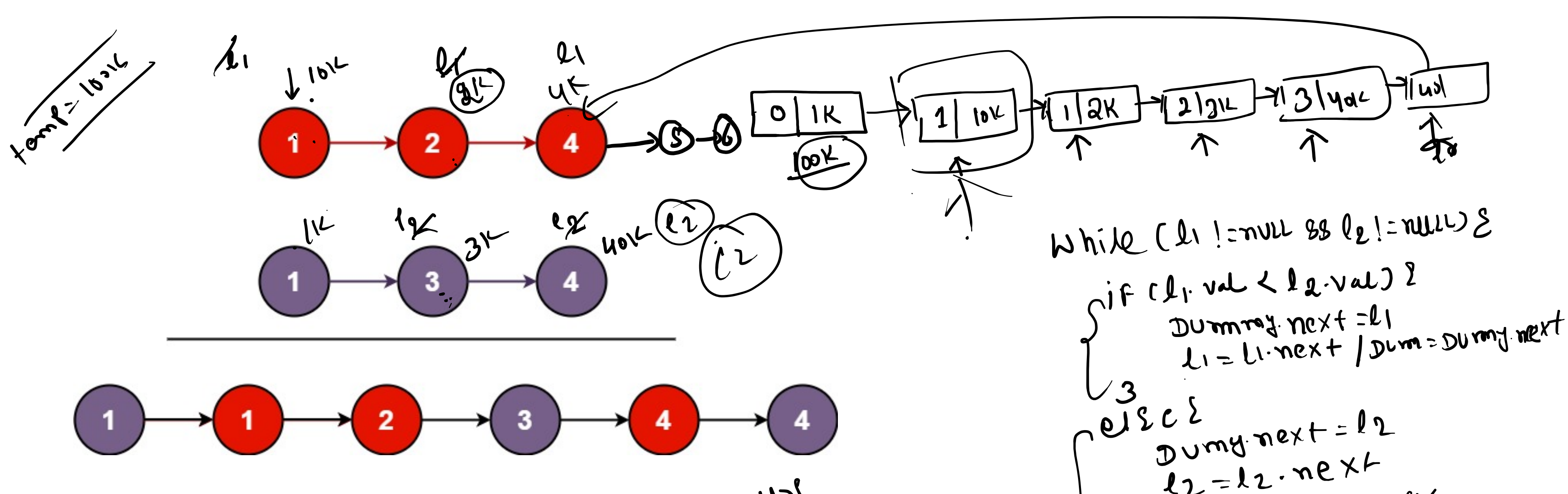
curr = head ~~2K~~ 3K

prev = ~~null~~ ~~1K~~ 2K

T.C O(n)

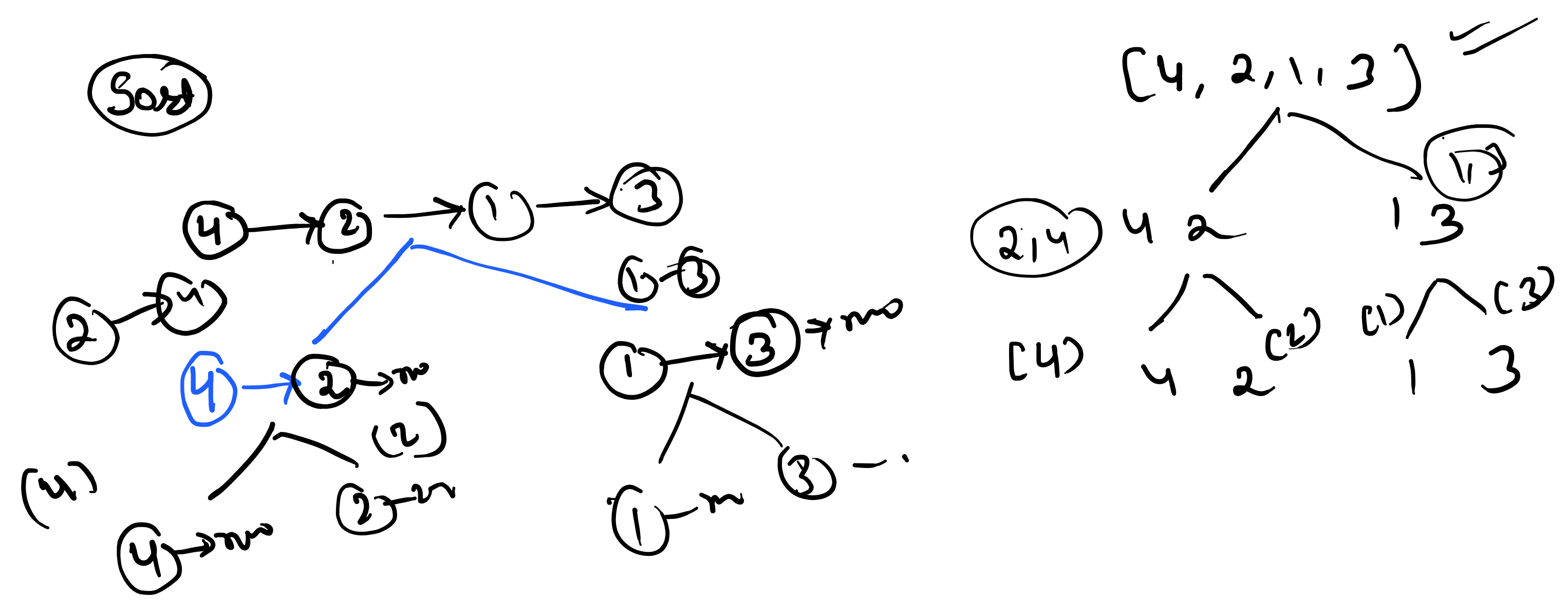
```
while(curr != null) {
    Node ahead = curr.next
    curr.next = prev
    prev = curr
    curr = ahea
}
ret prev
```

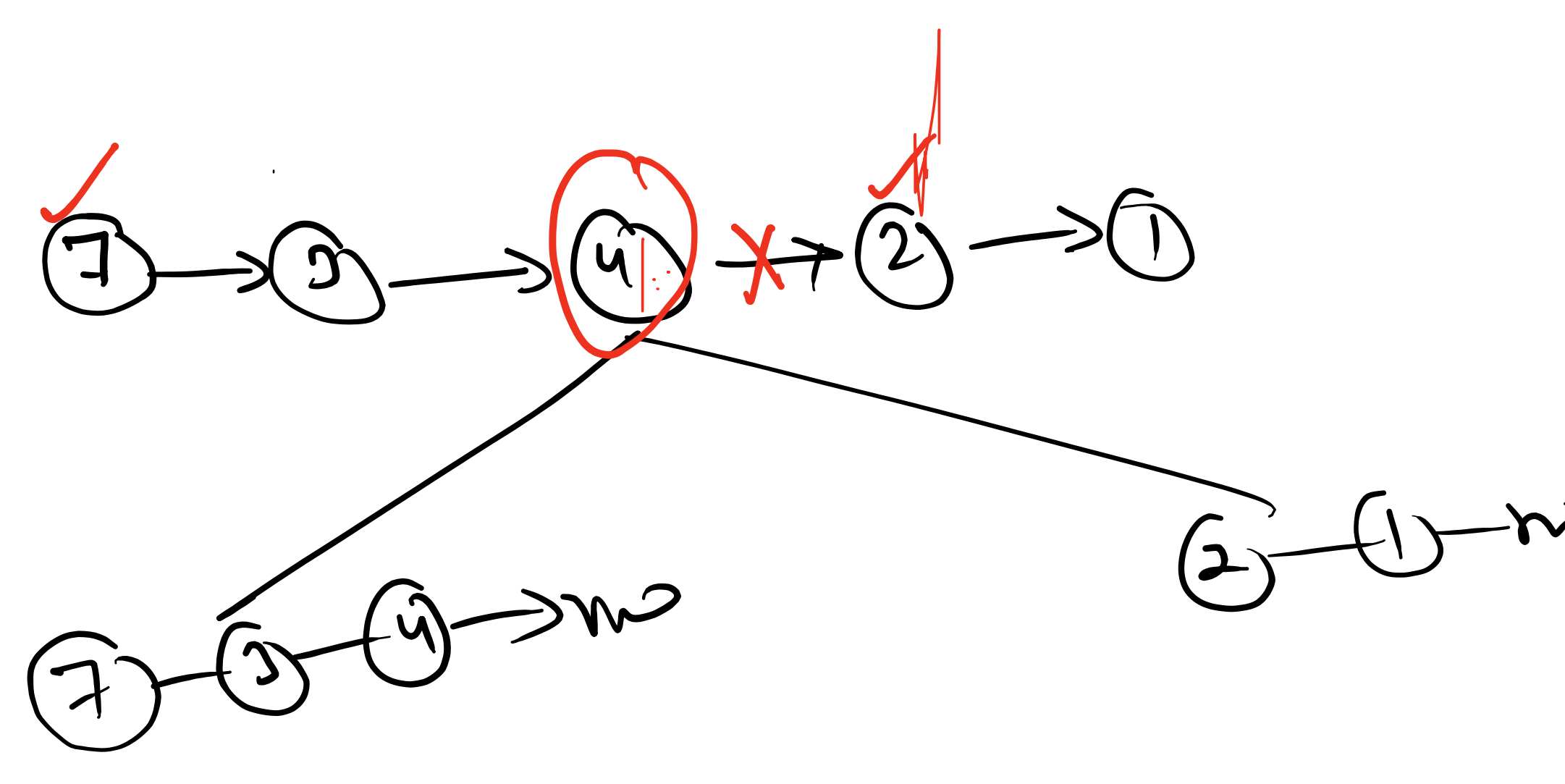


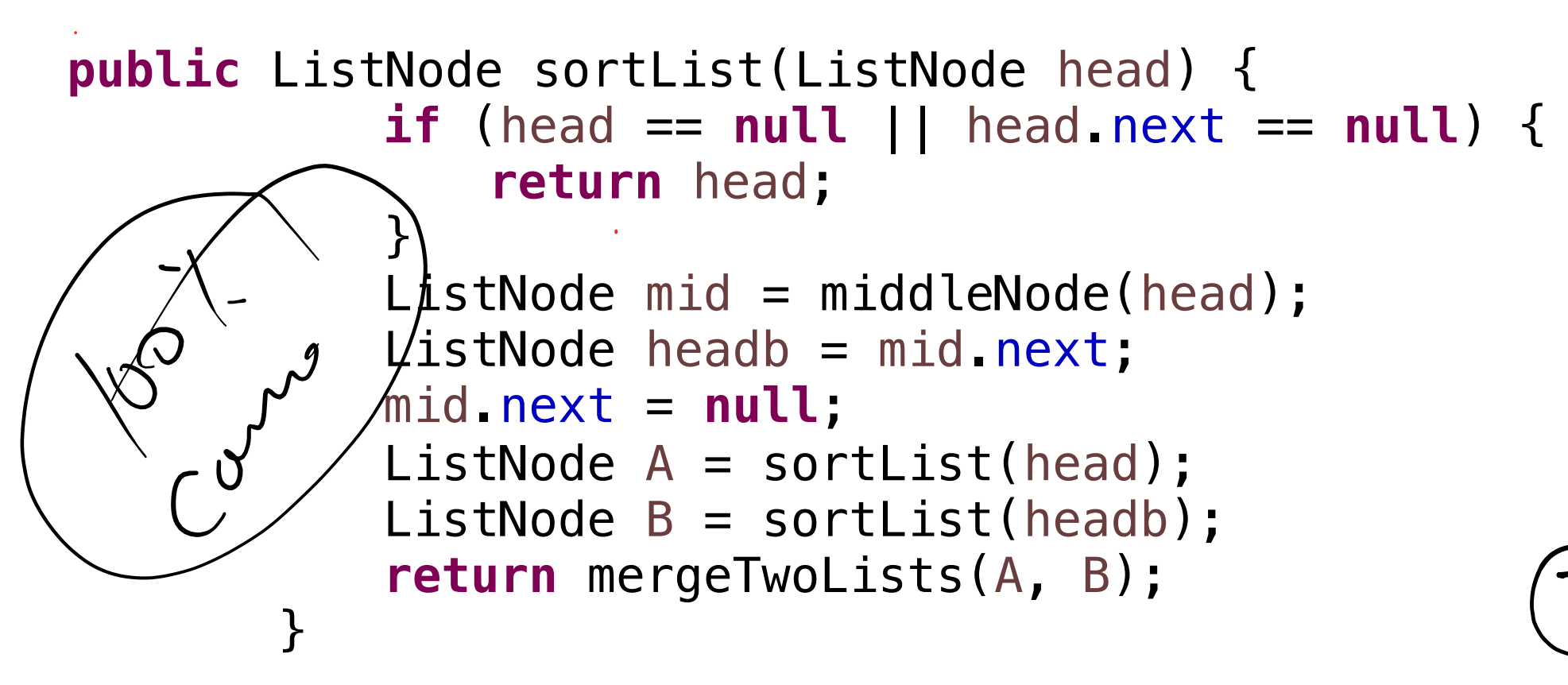


```
while (l1 != null && l2 != null) {
    if (l1.val < l2.val) {
        Dummy.next = l1
        l1 = l1.next / Dum = Dummy.next
    }
    else {
        Dumy.next = l2
        l2 = l2.next
        Dumy = Dumy.next
    }
}
```

```
if (l1 != null) {
    Dumm.next = l1
}
if (l2 != null) {
    Dumm.n = l2
}
temp.ne
```

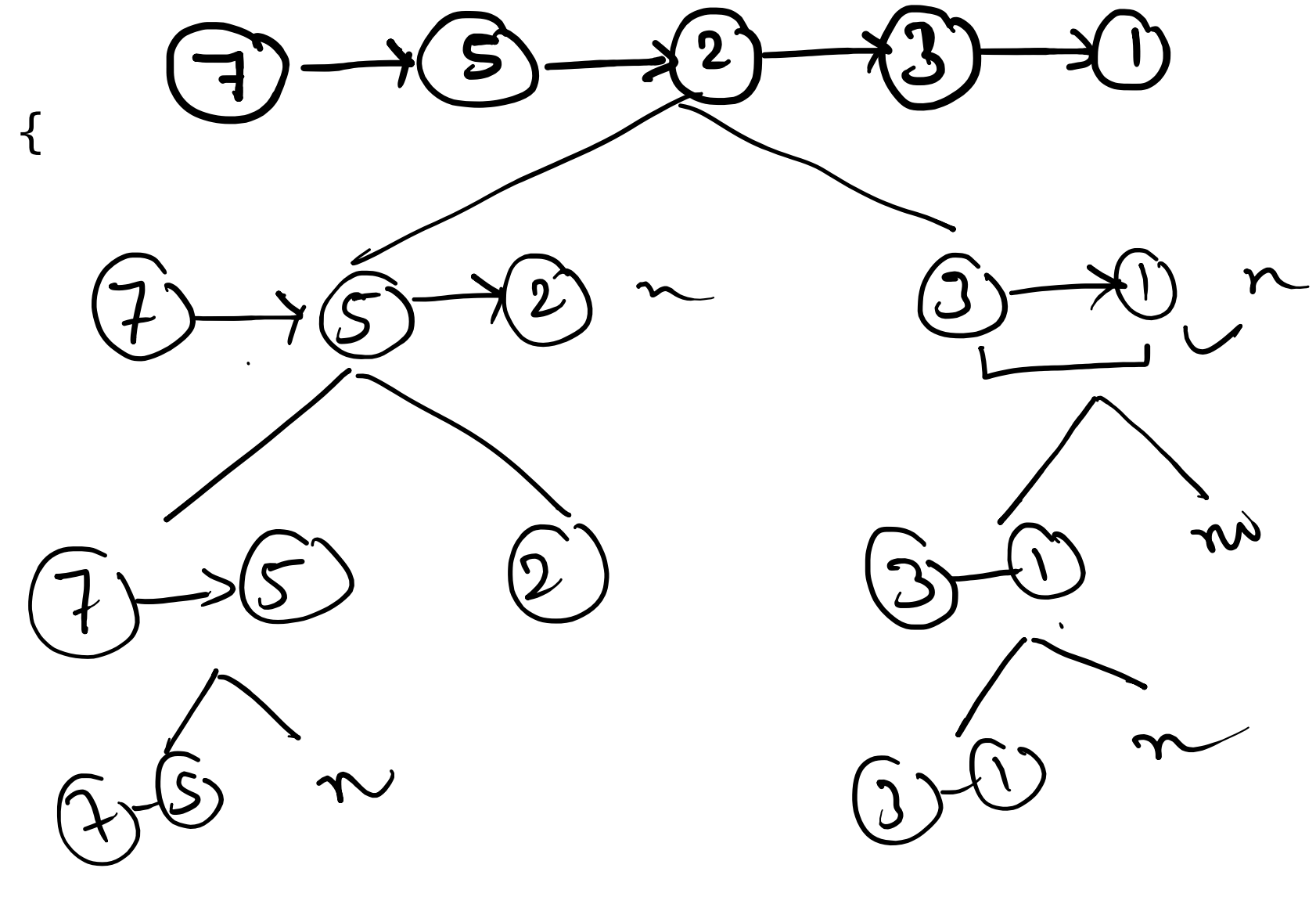




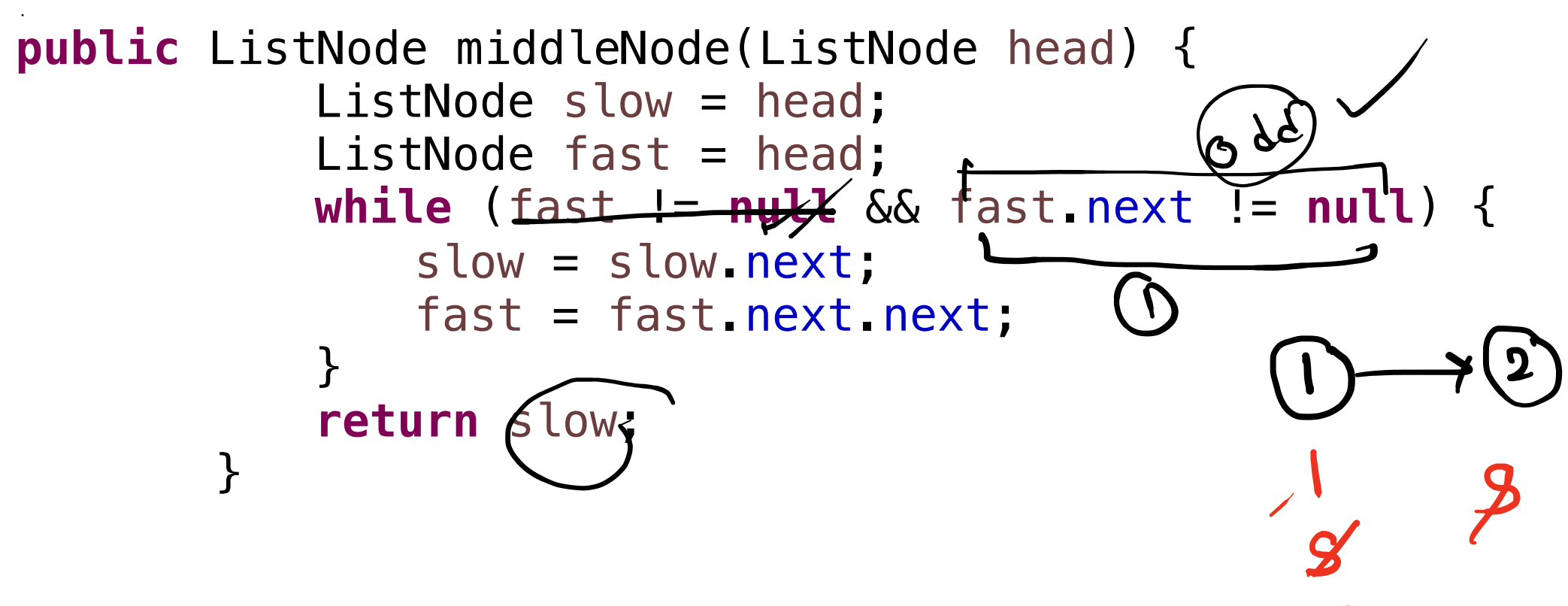


100% Come

```
public ListNode sortList(ListNode head) {
        if (head == null || head.next == null) {
            return head;
        }
        ListNode mid = middleNode(head);
        ListNode headb = mid.next;
        mid.next = null;
        ListNode A = sortList(head);
        ListNode B = sortList(headb);
        return mergeTwoLists(A, B);
    }
```



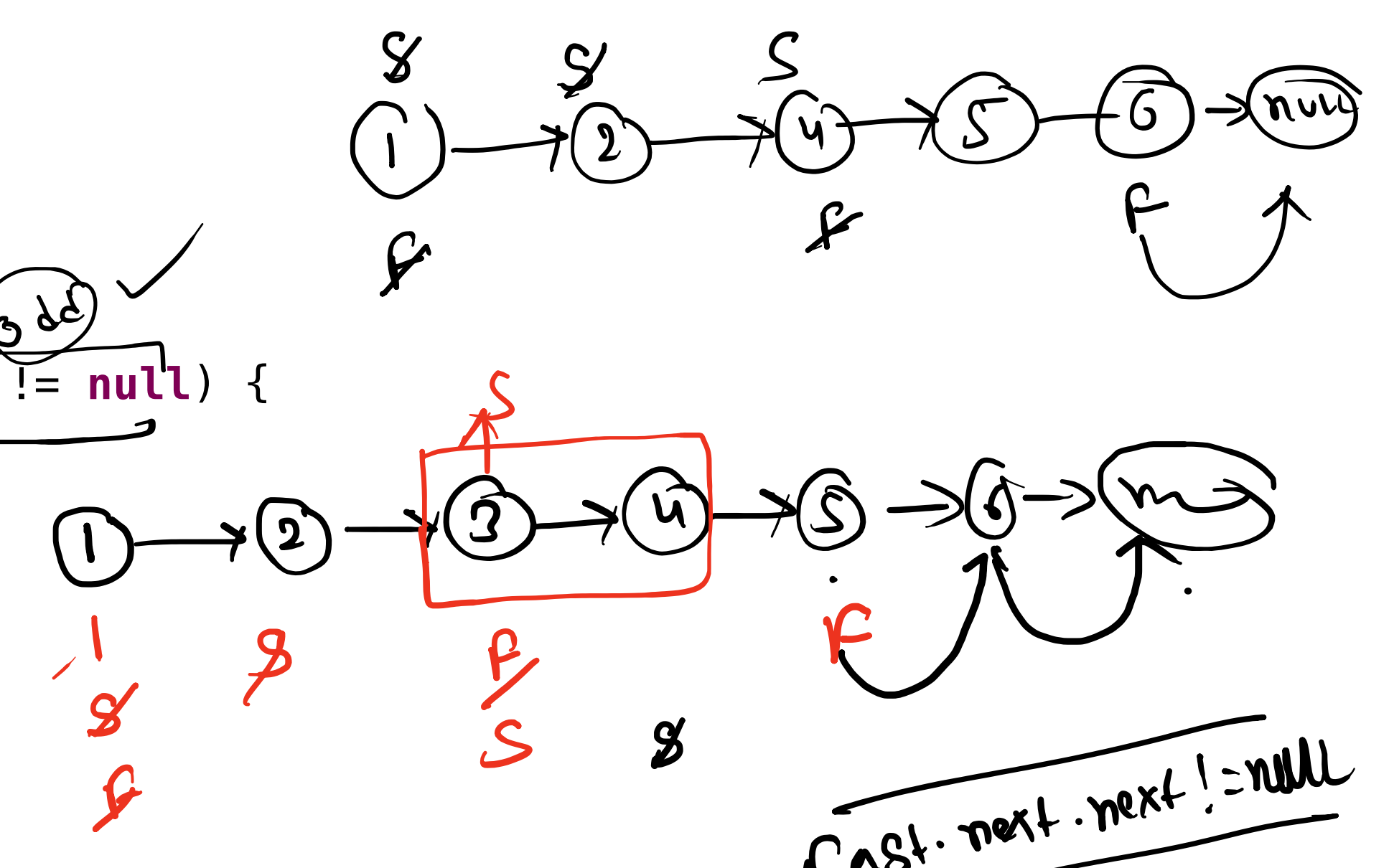
Floyed cycle



```
public ListNode middleNode(ListNode head) {
        ListNode slow = head;
        ListNode fast = head;
        while (~~fast != null~~ && fast.next != null) {
            slow = slow.next;
            fast = fast.next.next;
        }
        return slow;
    }
```

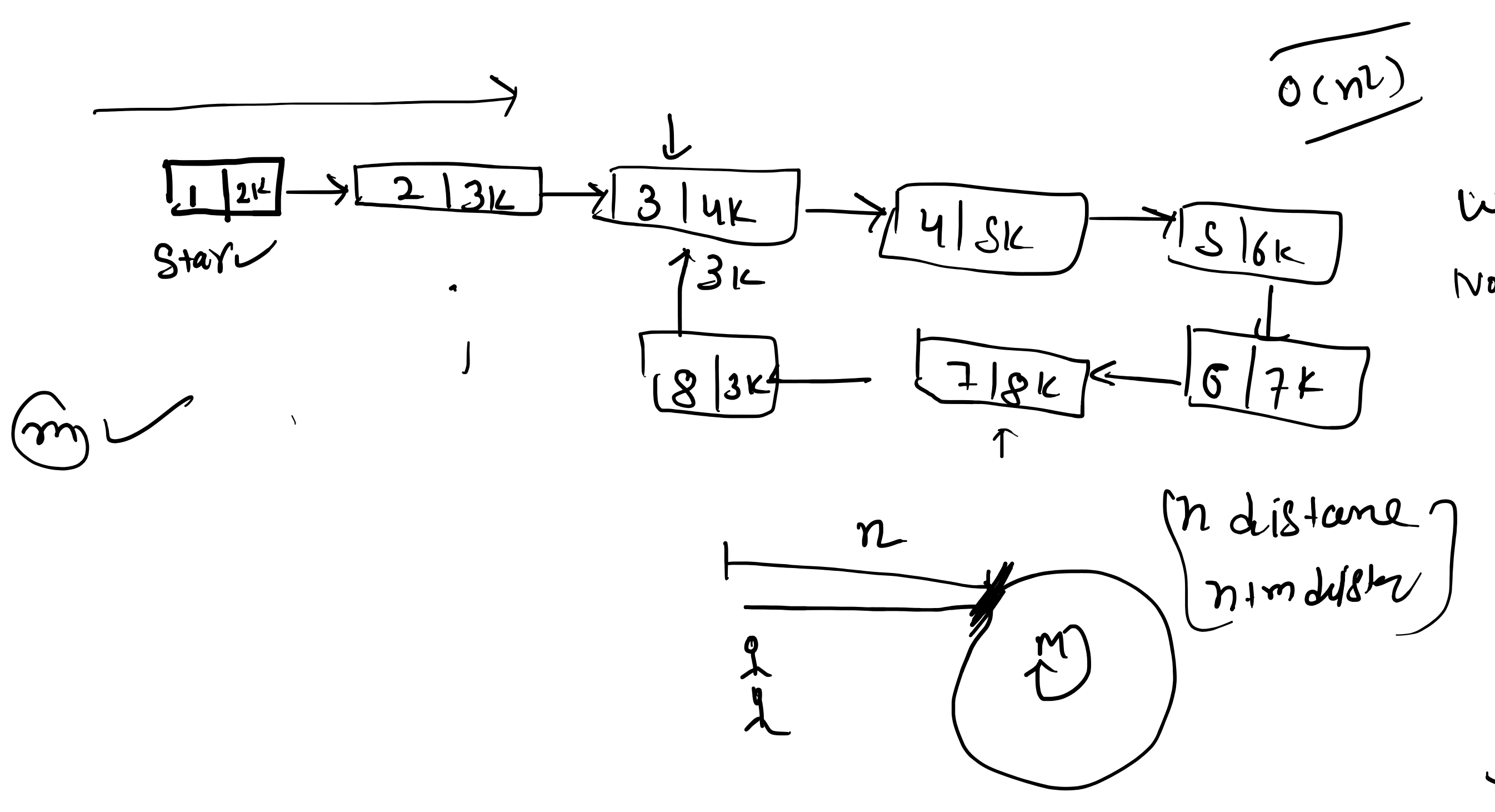
odd ✓

①



fast.next.next != null

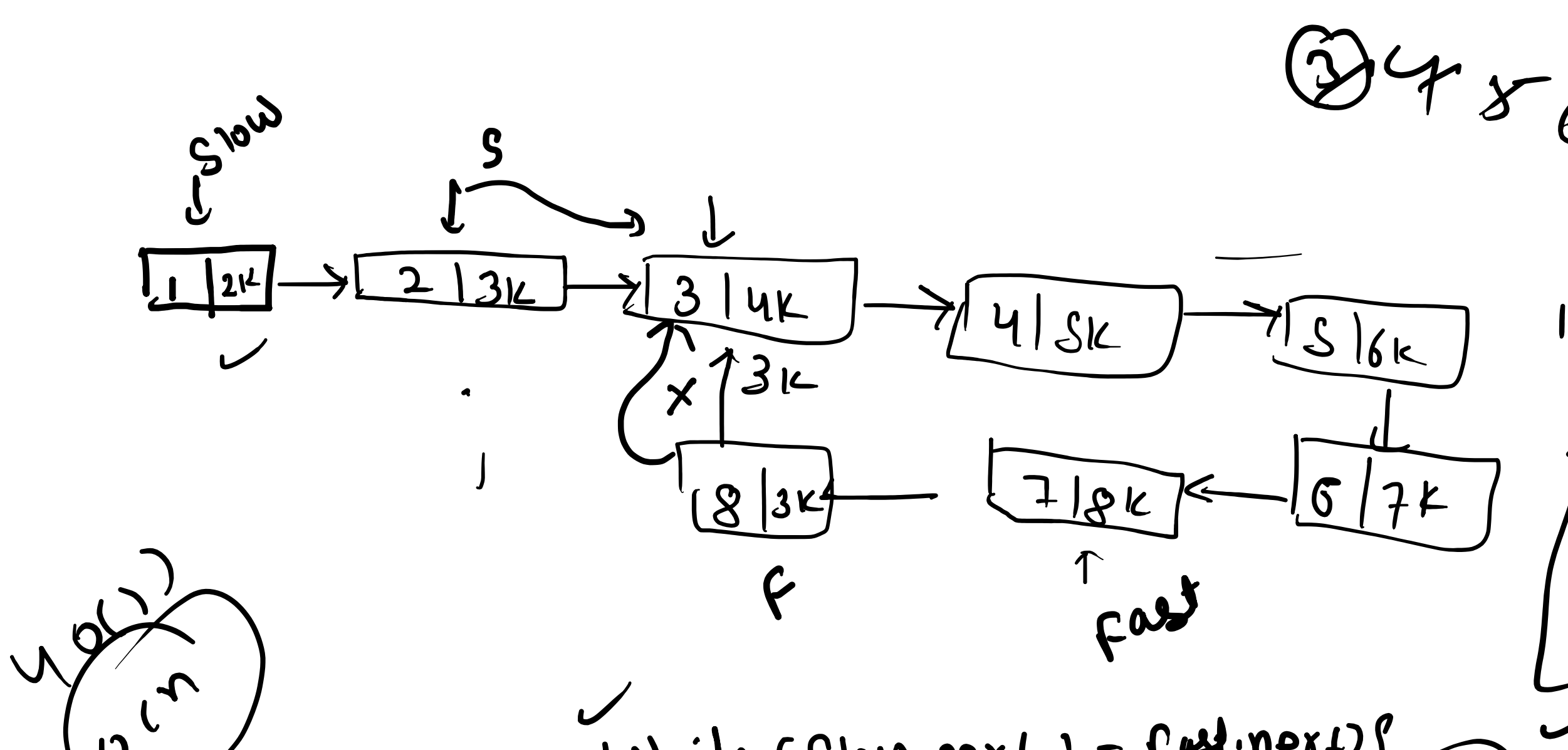
⑪



O(n²)

meet = 7K

```
while( start != null) {
    Node temp = meet
    while(temp.next != meet) {
        if(temp.next == start) {
            temp.next = null
            return
        }
        temp = temp.next
    }
    start = start.next
}
```



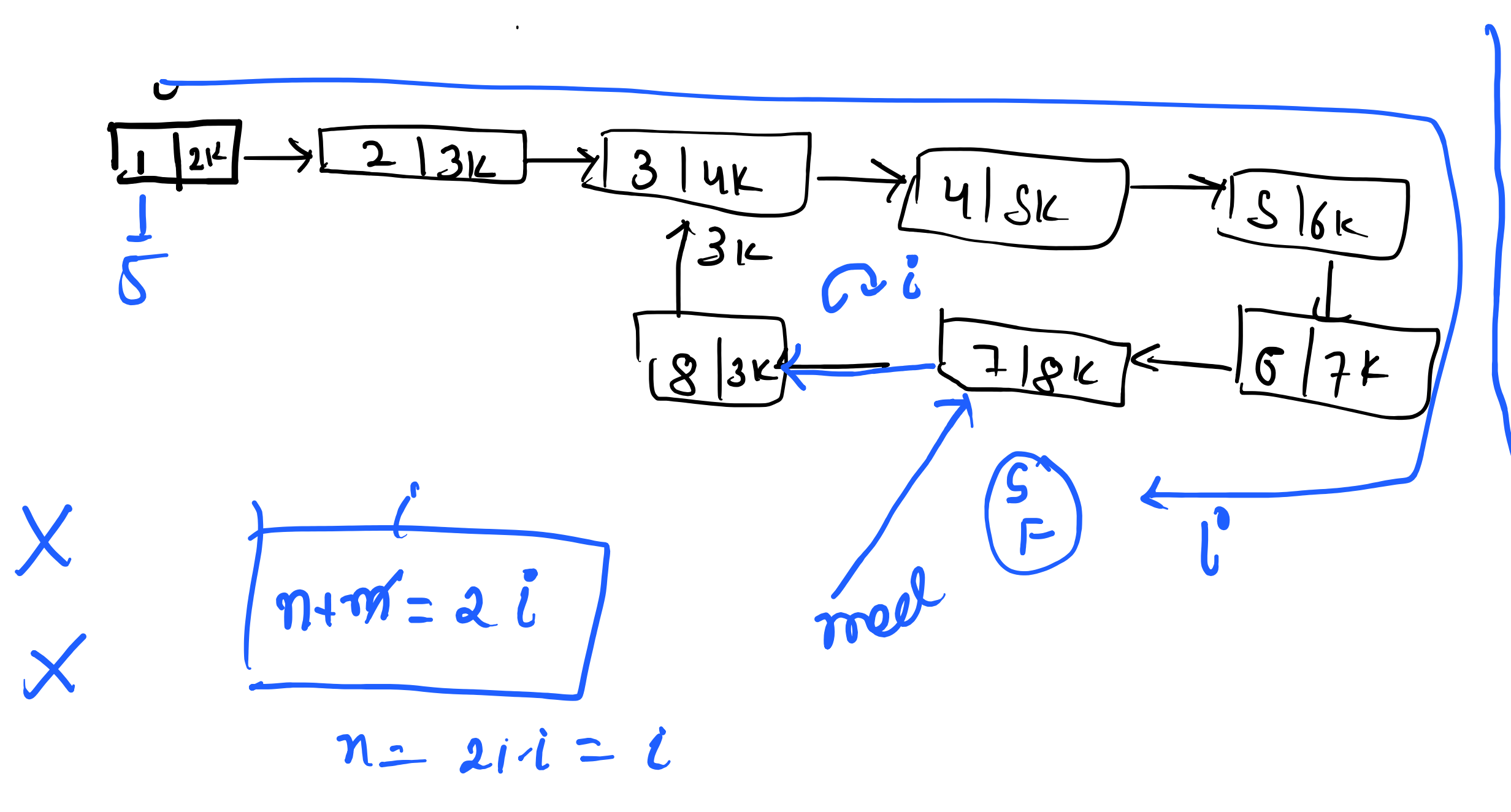
3 4 5 6

int c=1

```
while(slow.next != fast.next) {
    slow = slow.next
    fast = fast.next
}
fast ✓
```

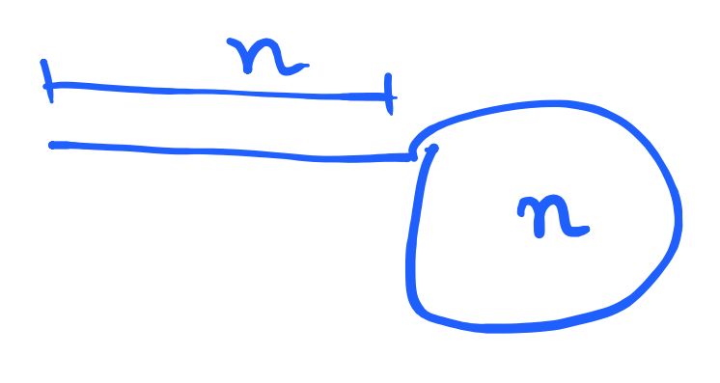
```
fast = head
Node temp = meet
while(temp.next != meet)
    c++
    temp = temp.next
}
```

```
for(i=0 i<c i++) {
    fast = fast.next
}
```



n+m = 2i

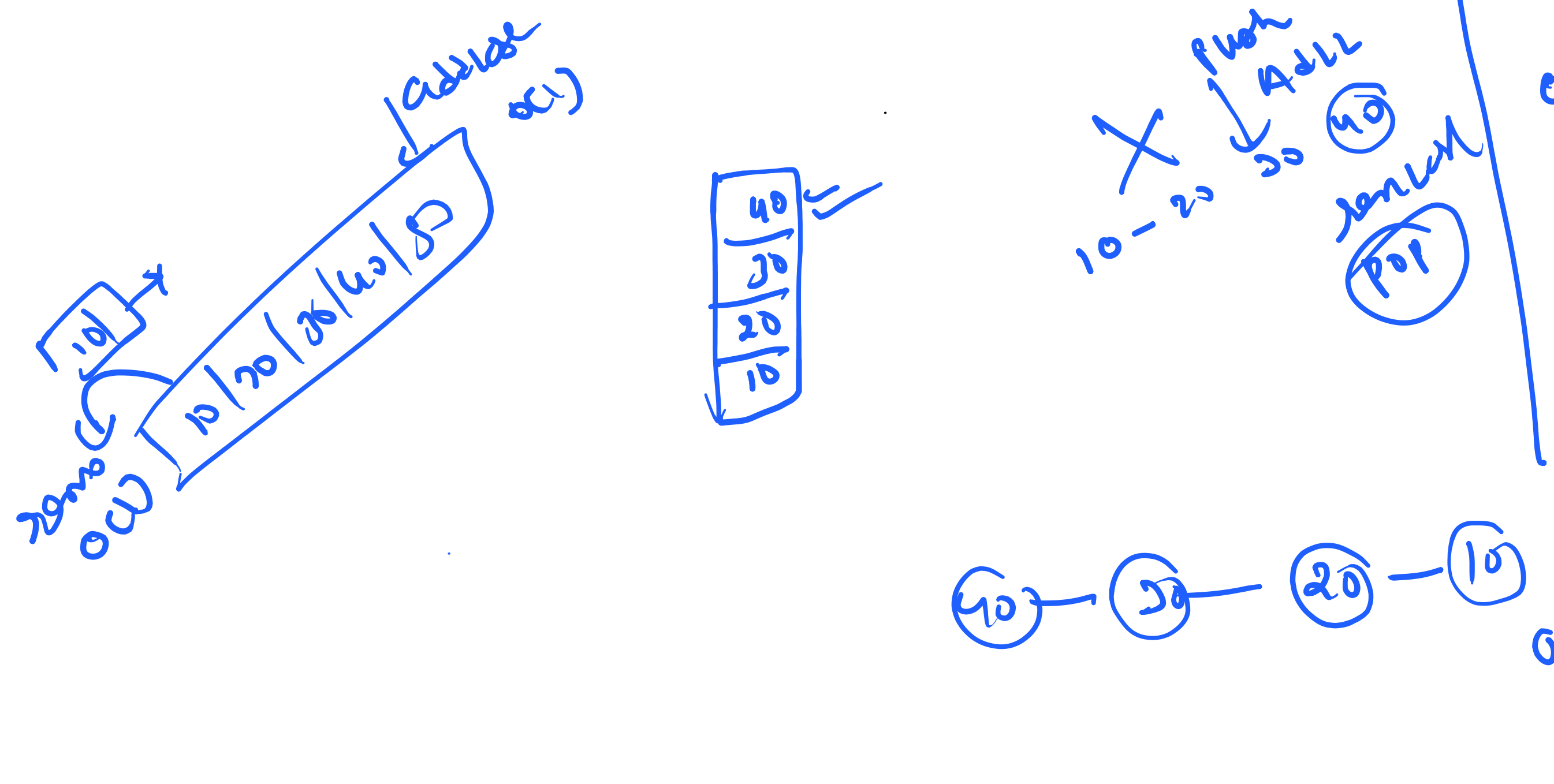
n = 2i-i = i



n = n+m

s = i

f → 2×i



O(n)

O(n)

O(1)

Add first → push

remove first → pop

Queue → Add last remove first ✓