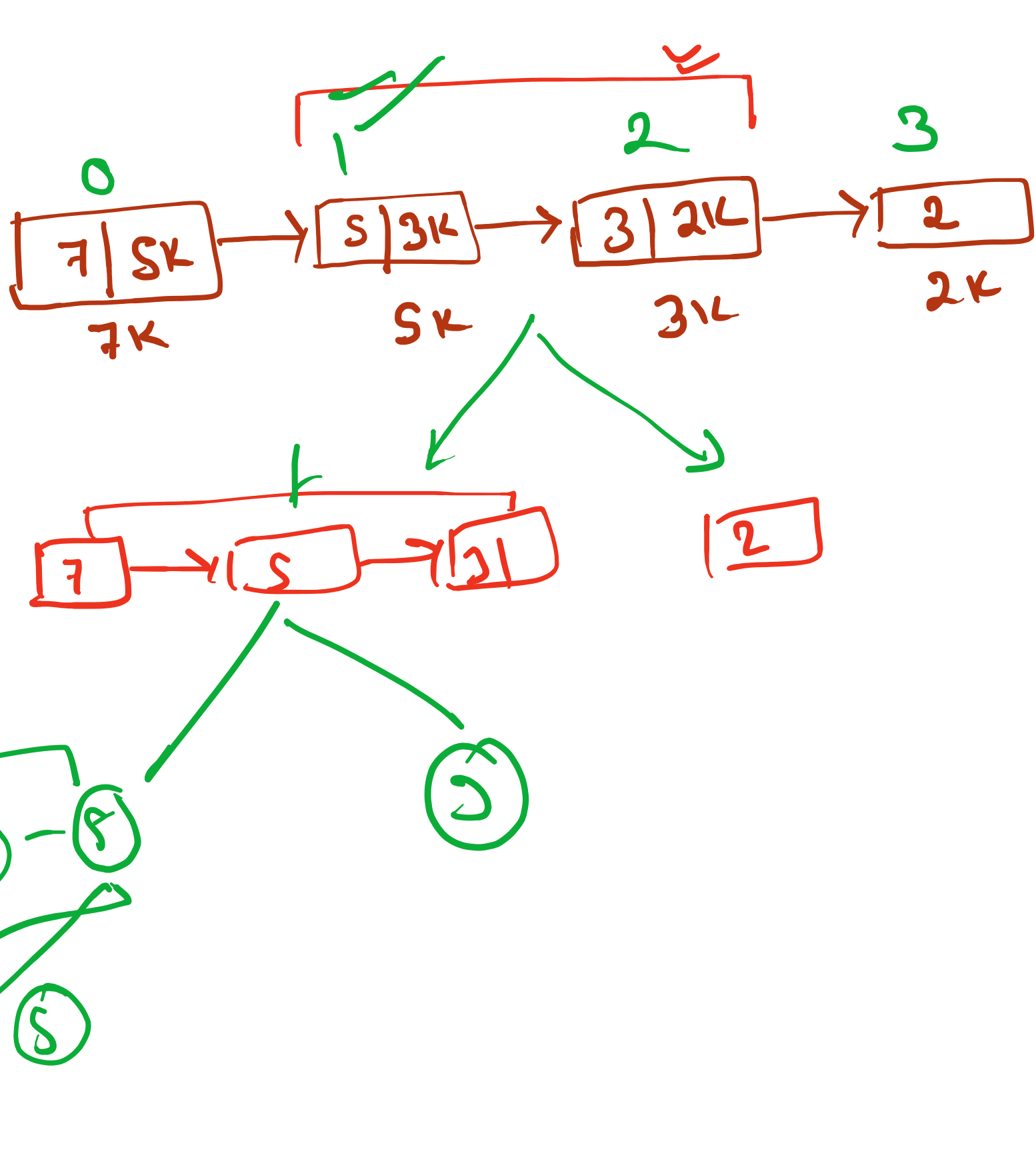
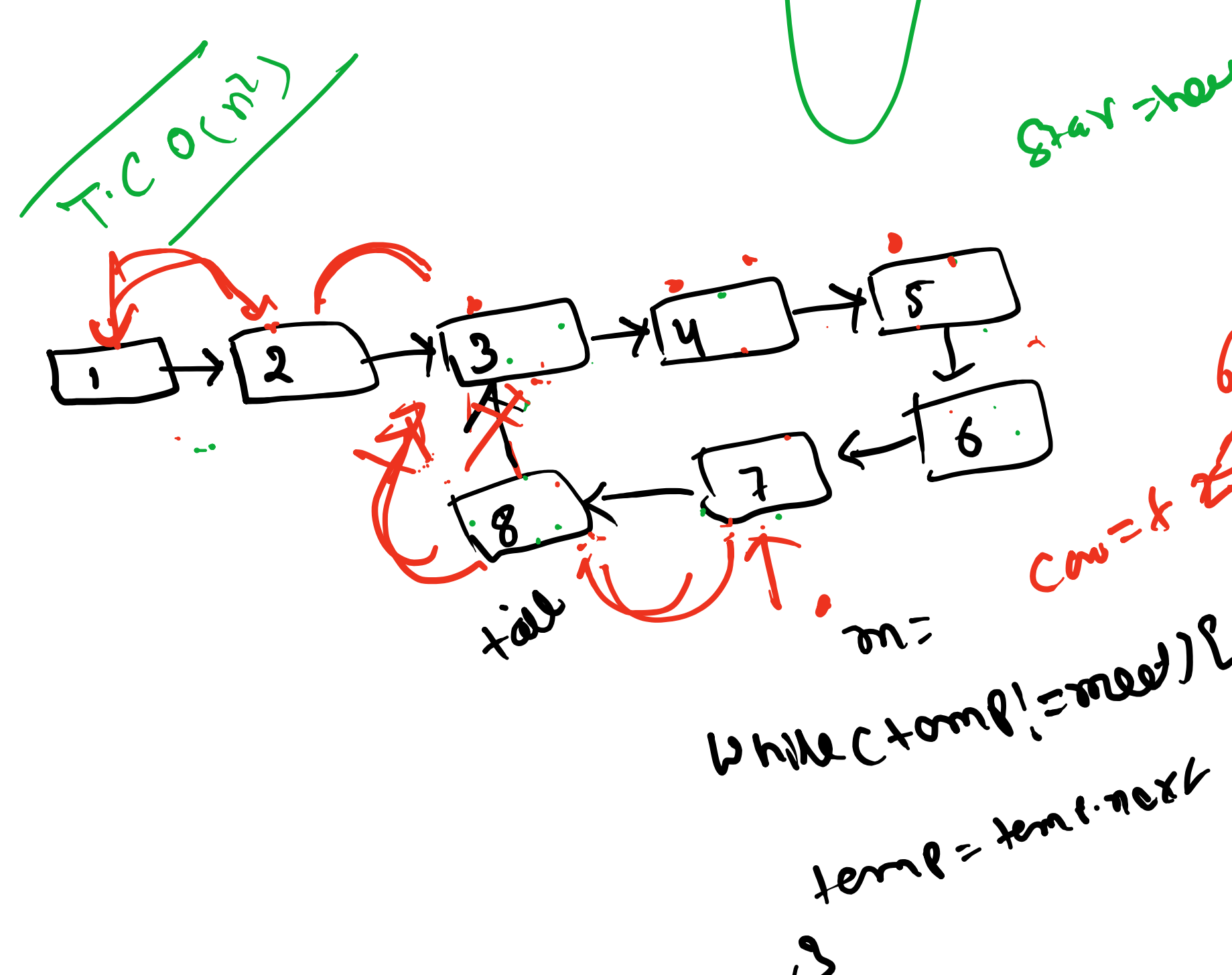
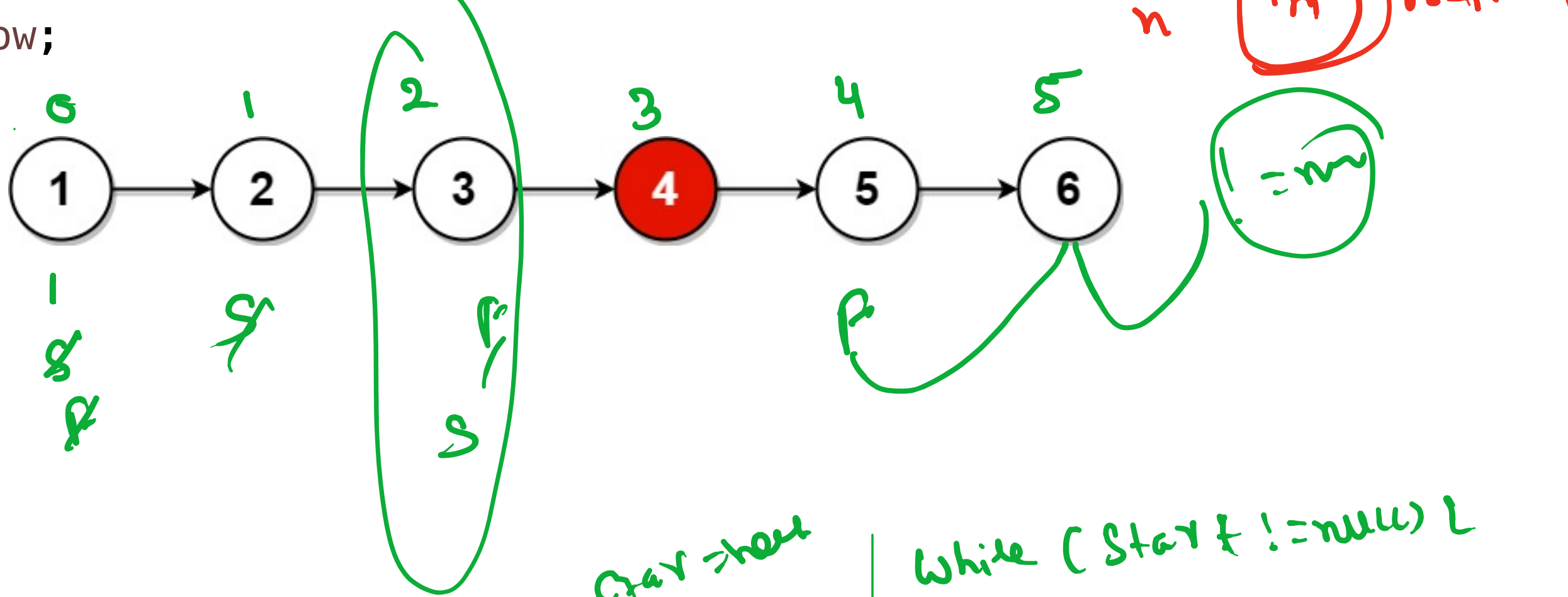


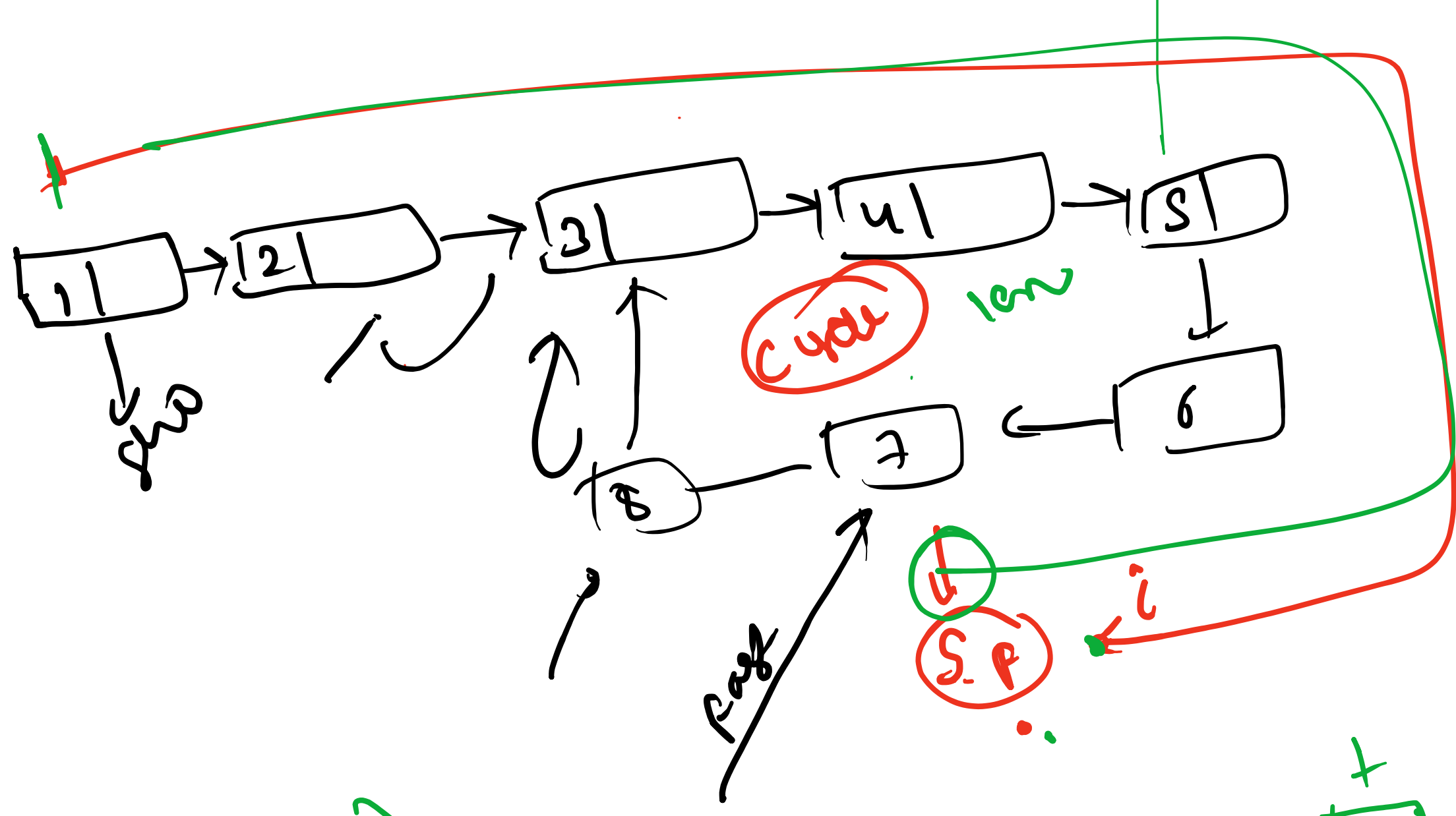
```
public ListNode sortList(ListNode head) {
    if(head==null || head.next==null) {
        return head;
    }
    ListNode mid = middleNode(head);
    ListNode headb = mid.next;
    mid.next = null;
    ListNode A = sortList(head);
    ListNode B = sortList(headb);
    return mergeTwoLists(A, B);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



```
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;
}
```



```
while (start != null) {
    Node temp = start;
    while (temp.next != null) {
        if (temp.next == start) {
            temp.next = null;
            break;
        }
        temp = temp.next;
    }
    start = start.next;
}
```

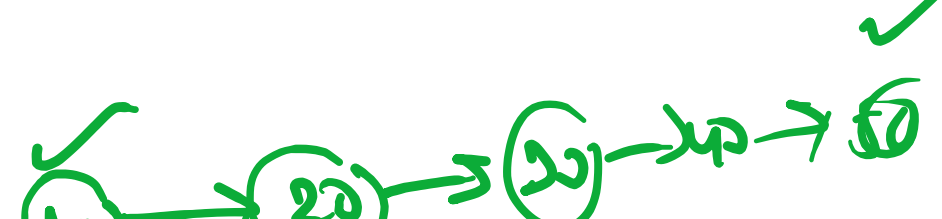
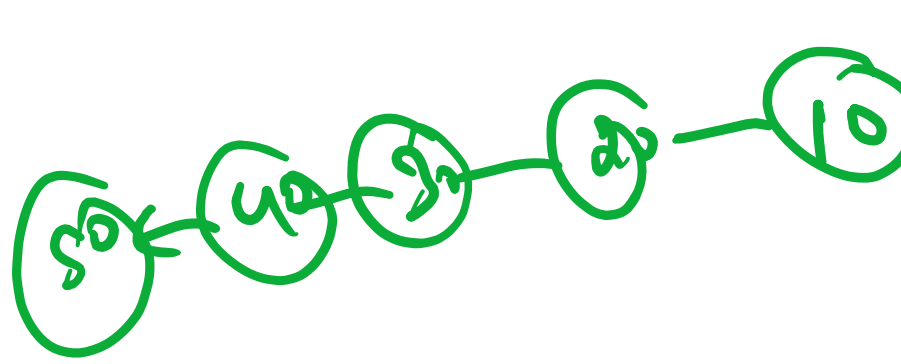


Slow = i distance
fast = 2xc

add links
down list

0(n) of add links
0(n) of remove links

0(n) of address
removing



add first -> 0(n)
removing 0(n)