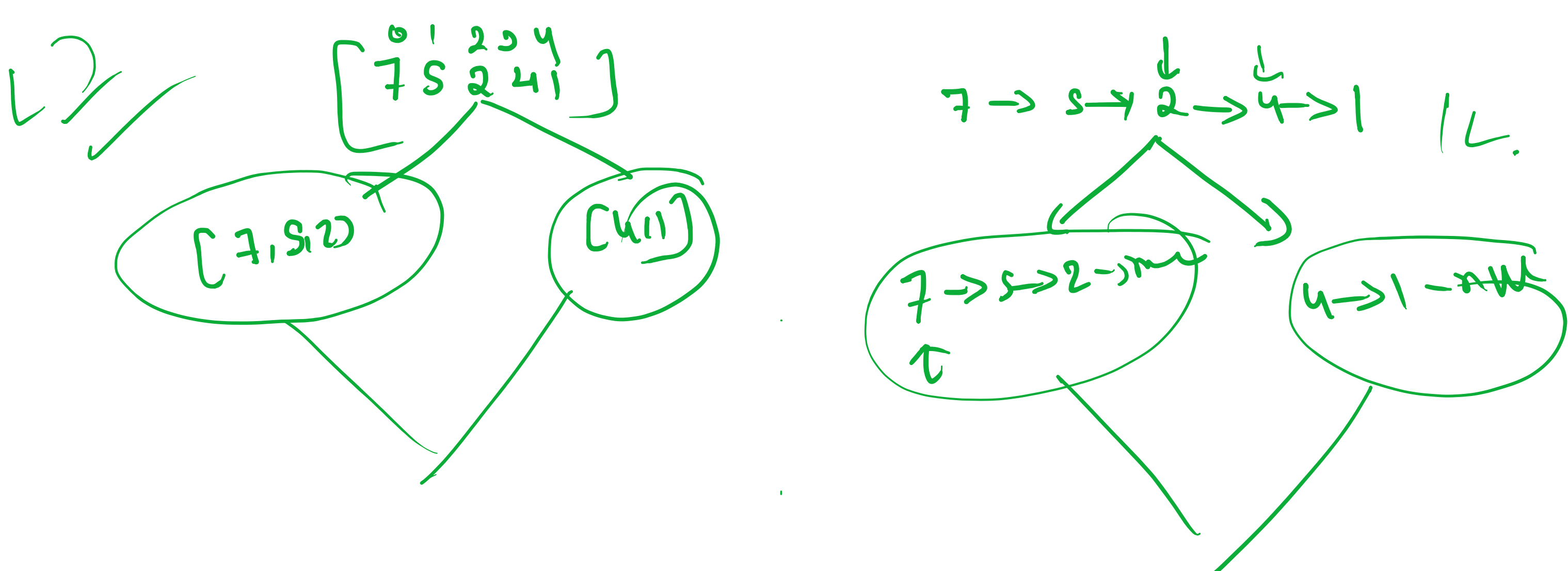
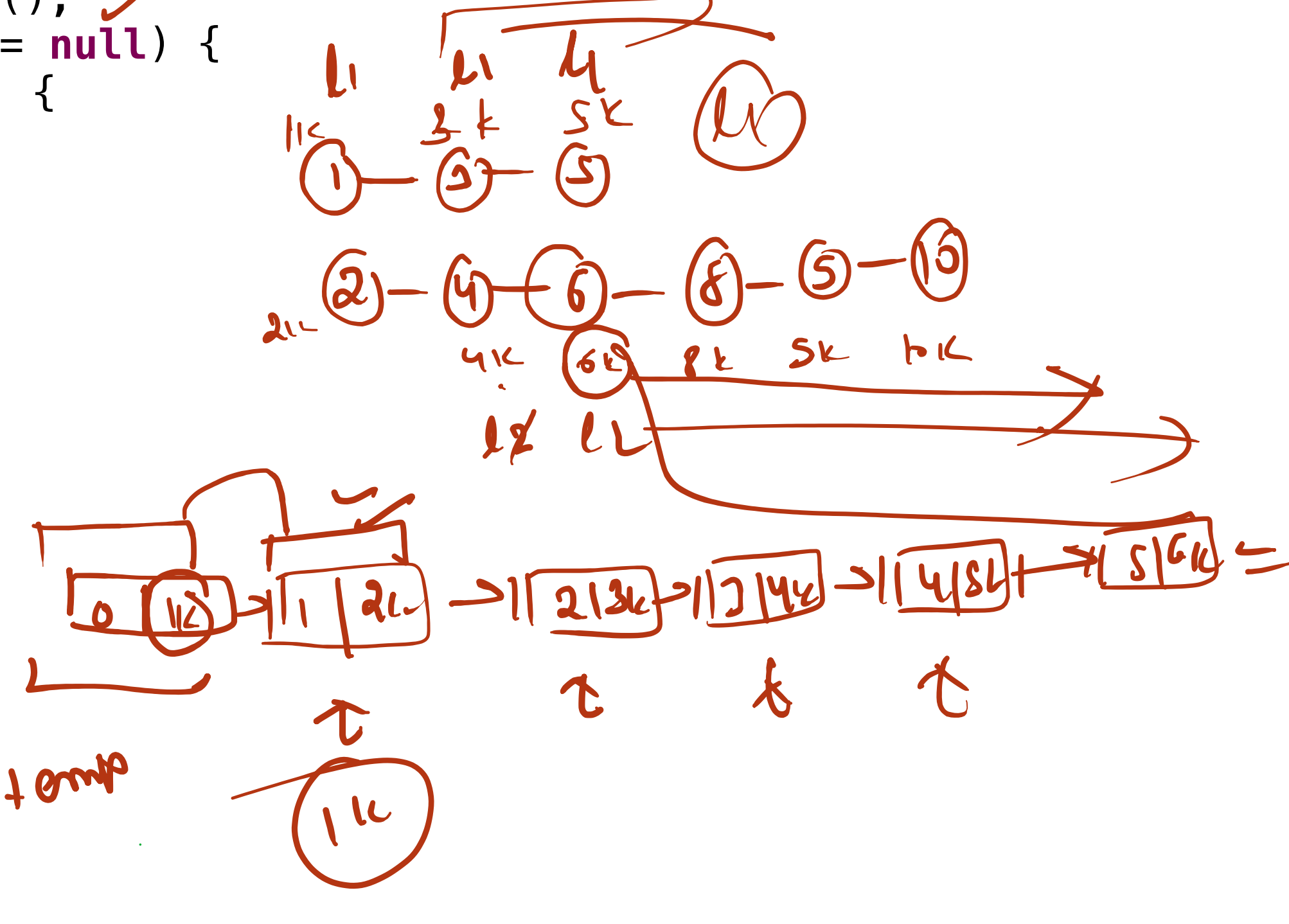
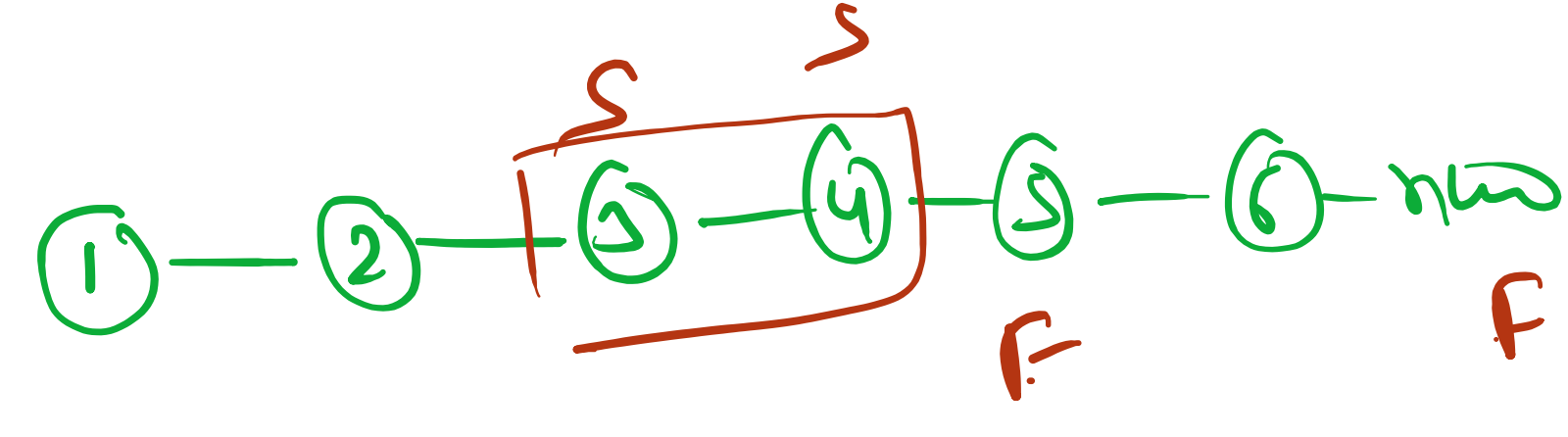
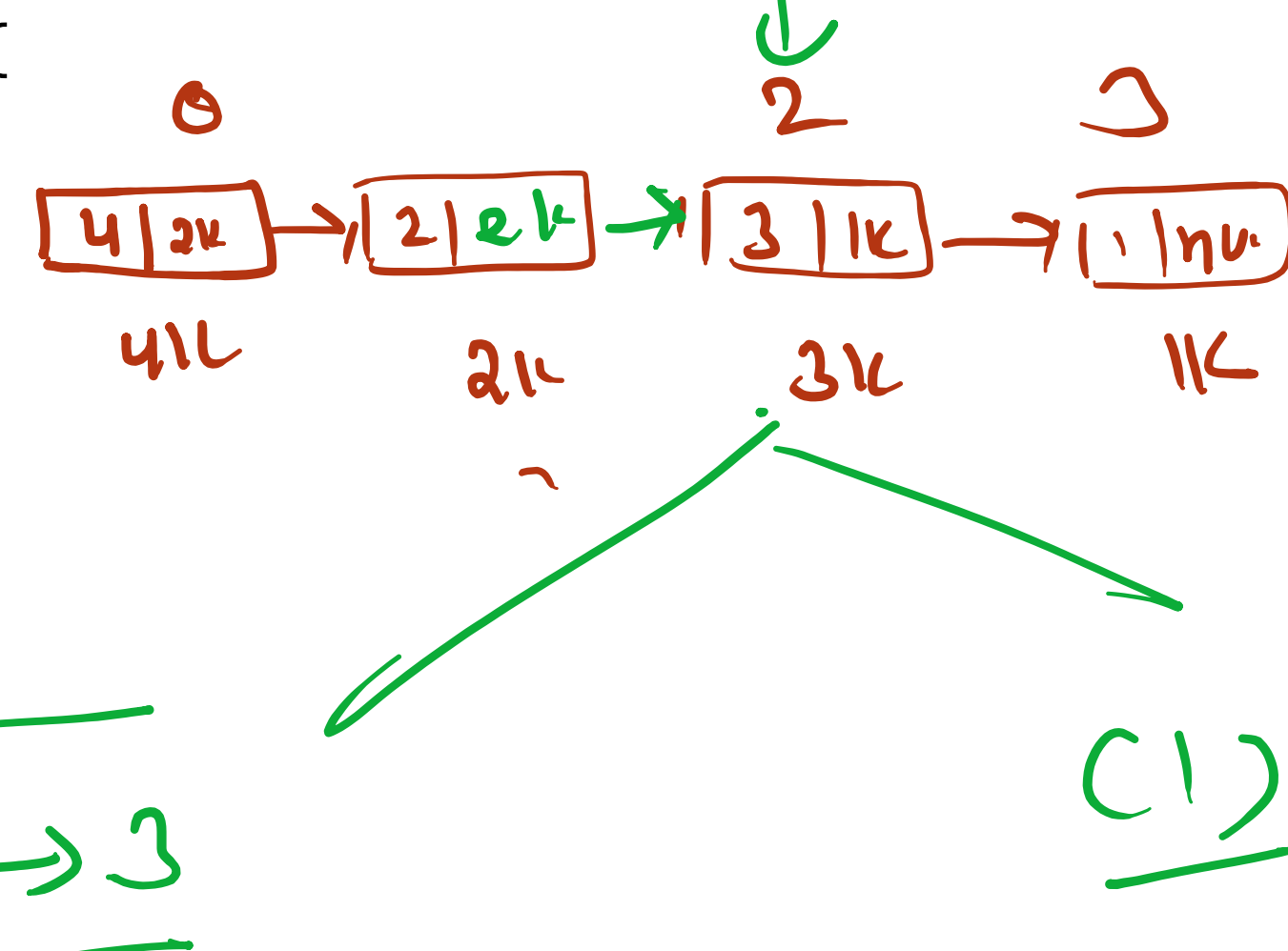


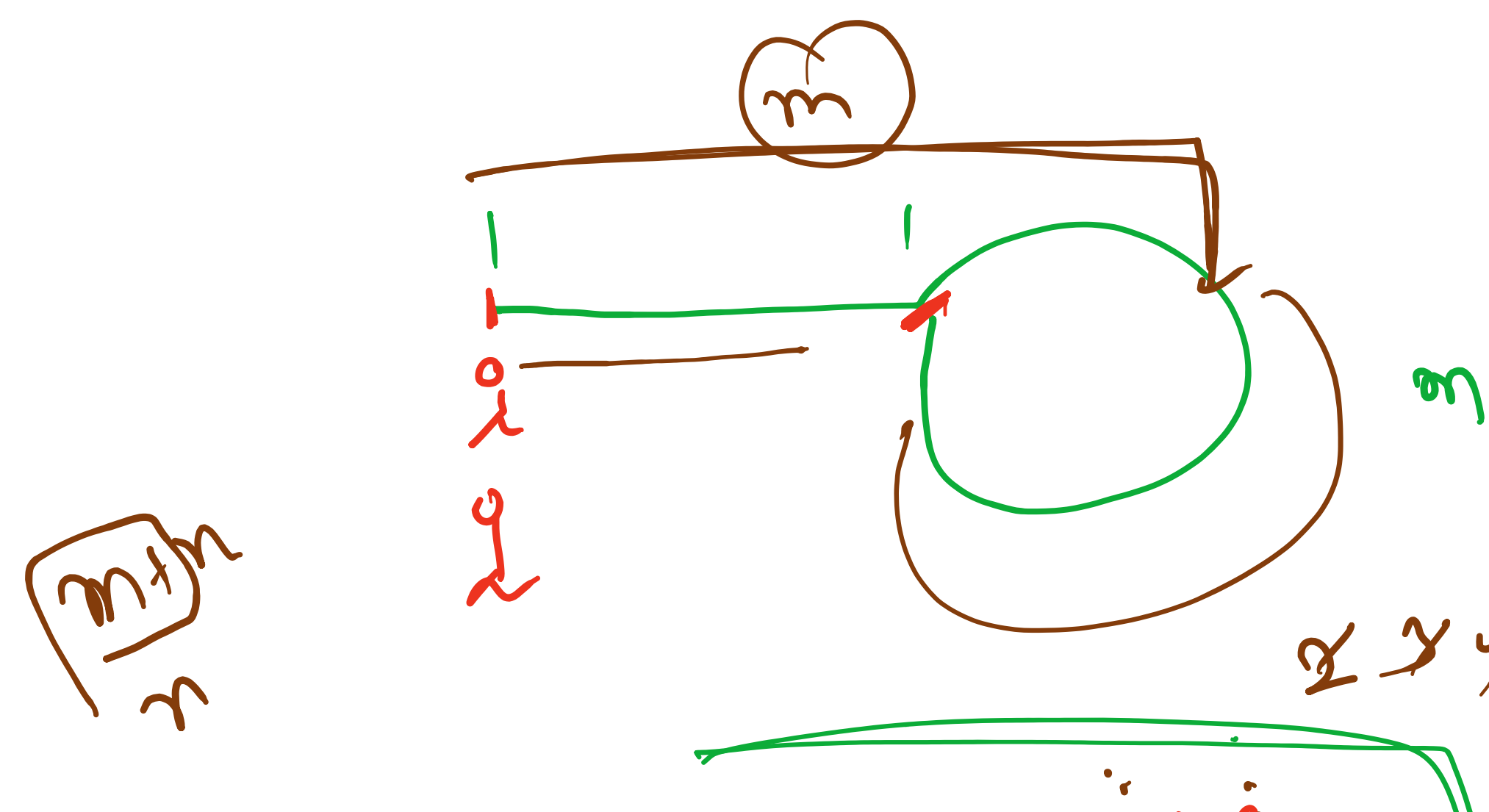
```
public ListNode mergeTwoLists(ListNode list1, ListNode list2){
    ListNode dummy= newListNode();
    while(list1!= null&& list2!= null) {
        if(list1.val< list2.val) {
            dummy.next= list1;
            list1= list1.next;
            dummy= dummy.next;
        } else{
            dummy.next= list2;
            list2= list2.next;
            dummy= dummy.next;
        }
    }
    if(list1!= null) {
        dummy.next = list1;
    } else{
        dummy.next = list2;
    }
}
```



```
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);
}
```



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



```
Node slow= head;
while(slow.next!= fast.next) {
    slow= slow.next;
    fast= fast.next;
}
fast.next= null;
```

```
1. cycle lena
int c=1;
Node temp= null;
while (temp.next!= null){
    c++;
    temp= temp.next;
}
```

addition -> new  
removal -> new  
update -> new

(0,1,2,3,4,5,6,7,8,9,10)



2. IFO