6

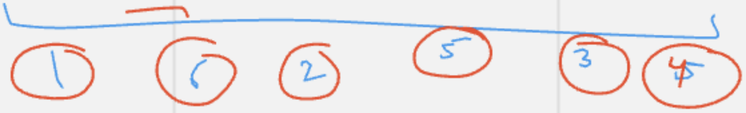
f L sf LS 3F L3

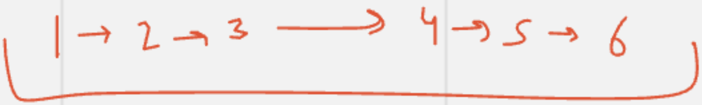
1 6 2 5 3 4

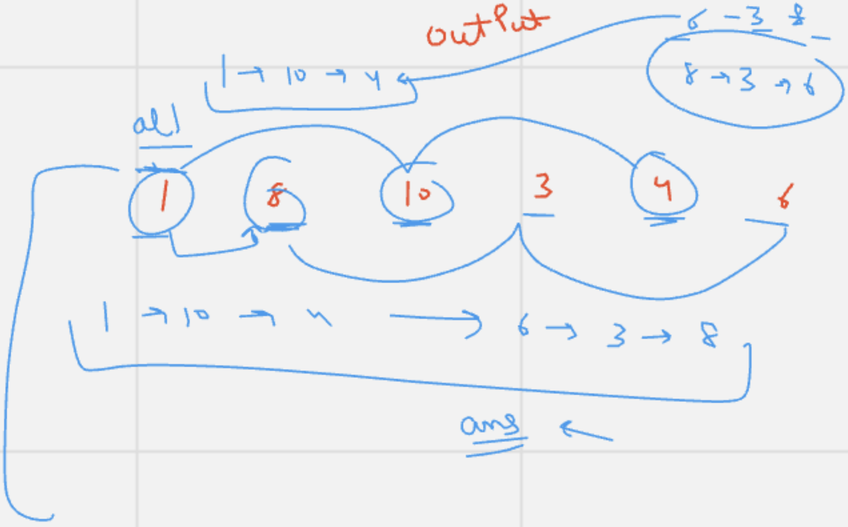
A linked list L0 -> L1 -> L2 -> ..... -> LN

find out

can be folded as L0 -> LN -> L1 -> LN – 1 -> L2 -> .....

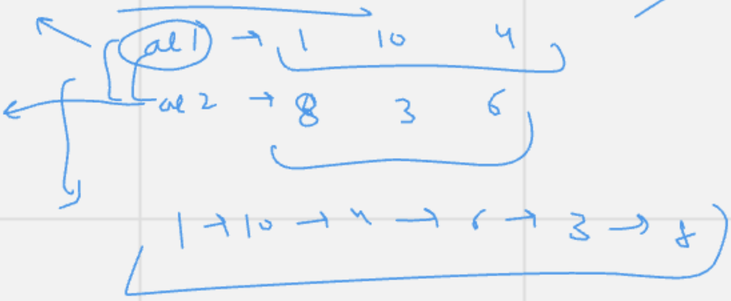






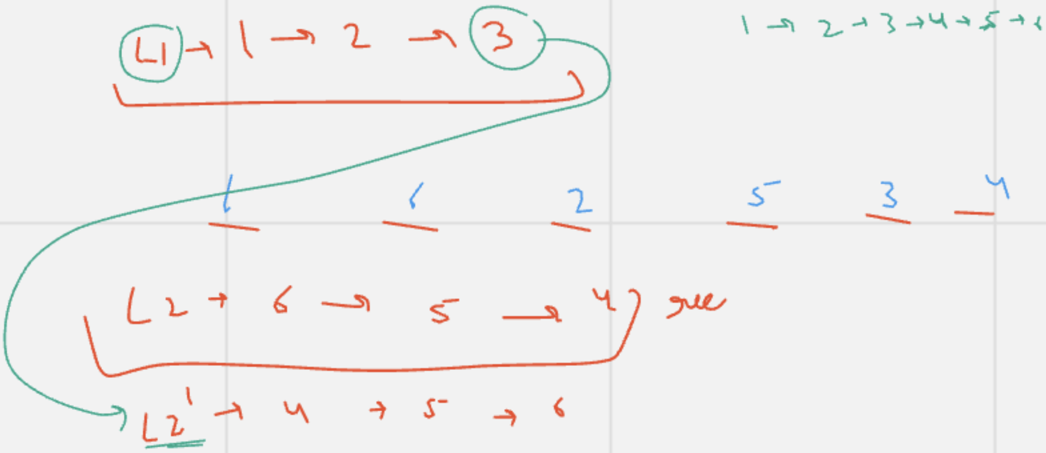
Solve this ??

Correct

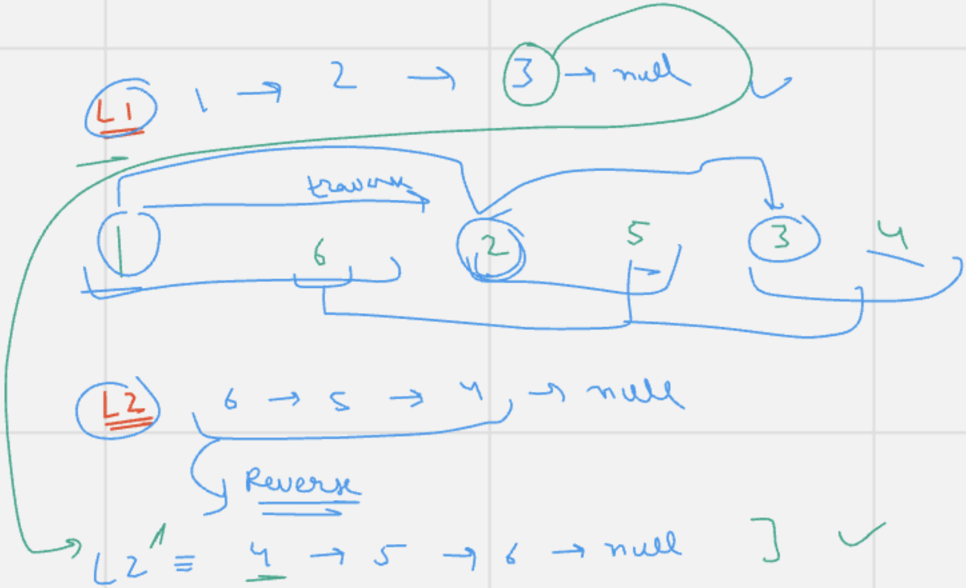


6

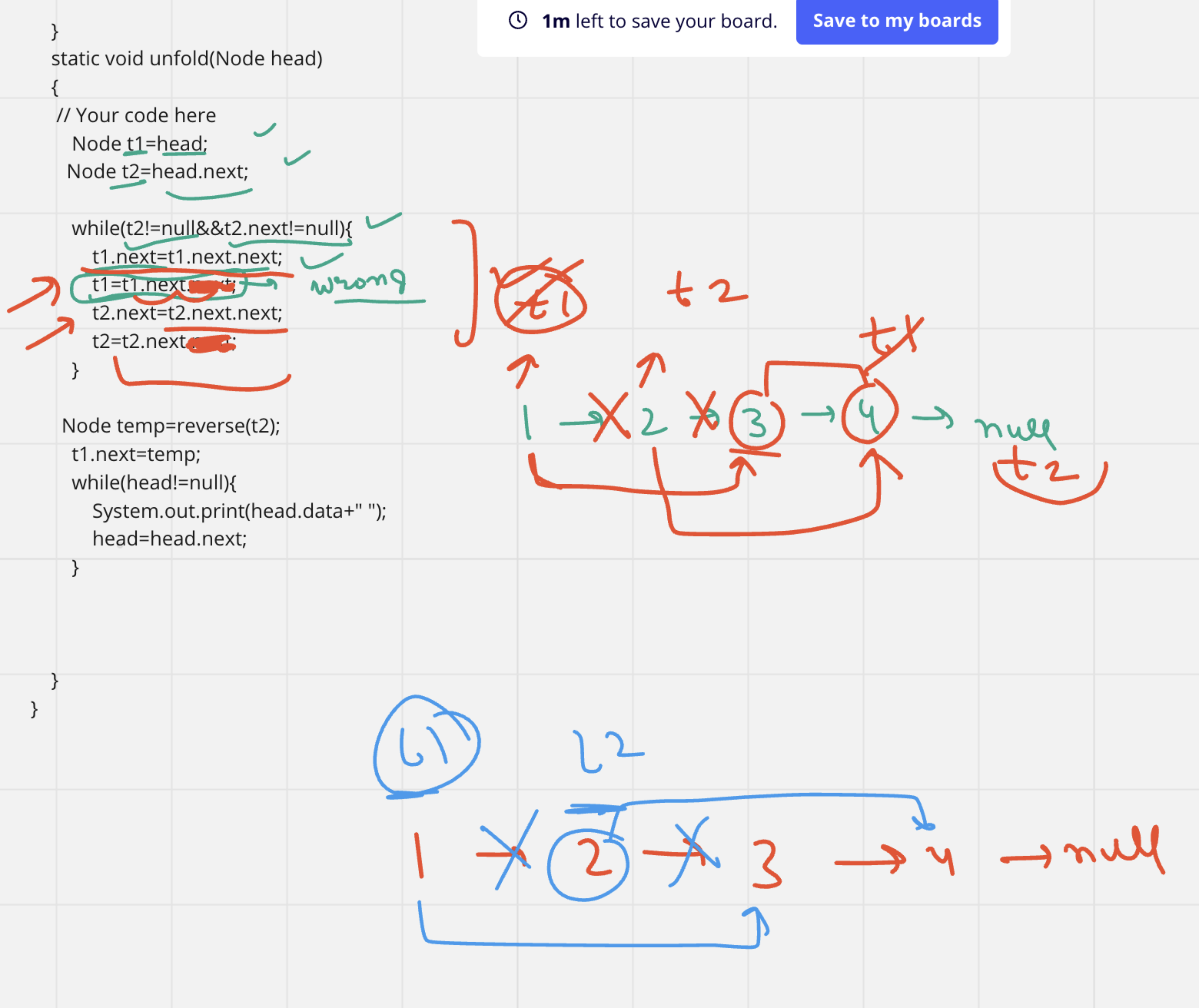
1 6 2 5 3 4



L1 → 1 → 2 → 3
1 → 2 → 3 → 4 → 5 → 6
1 6 2 5 3 4
L2 → 6 → 5 → 4 rev
L2' → 4 → 5 → 6



L1 1 → 2 → 3 → null
traverse
1 6 2 5 3 4
L2 6 → 5 → 4 → null
Reverse
L2' ≡ 4 → 5 → 6 → null



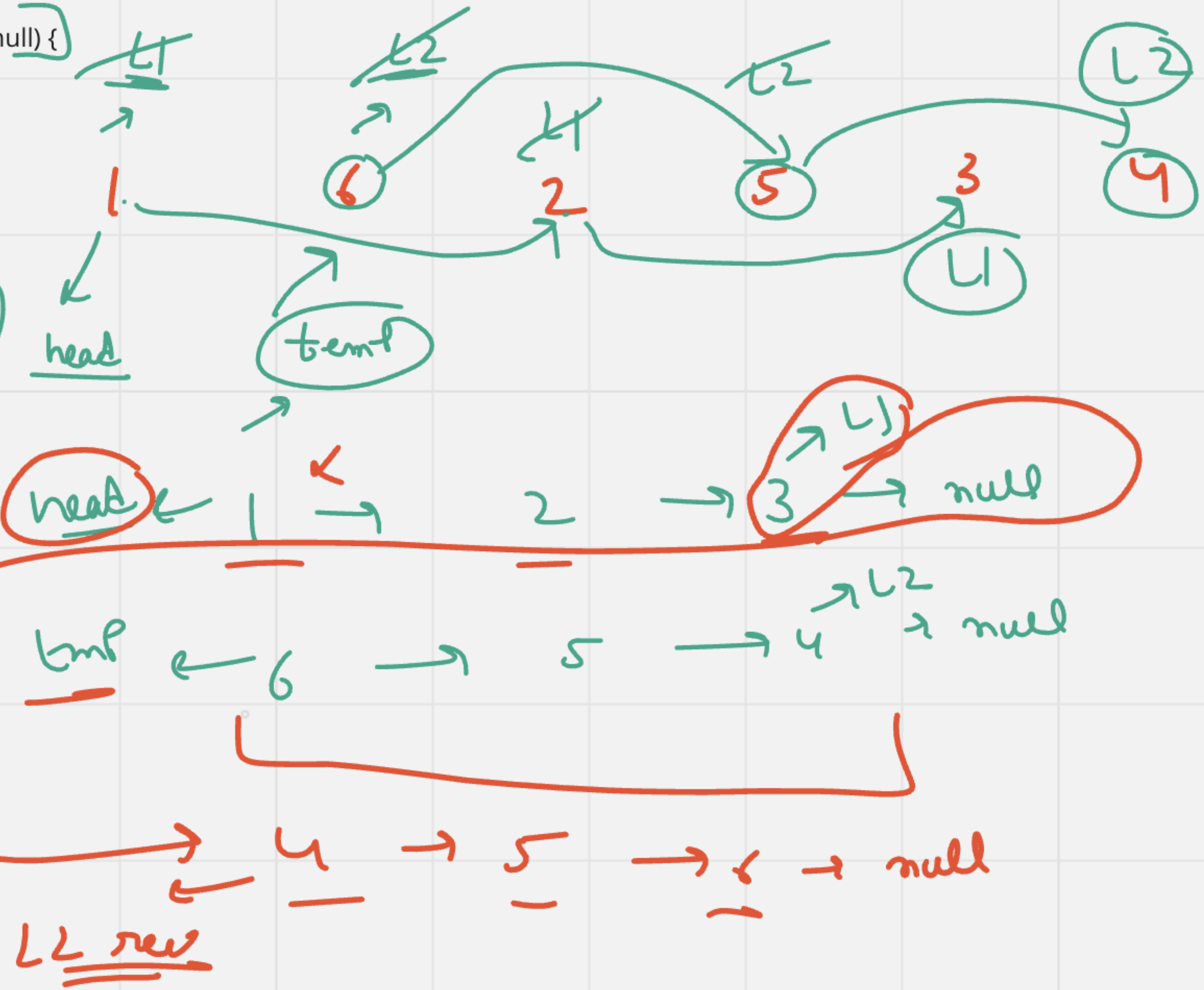
1m left to save your board. Save to my boards

```
}
static void unfold(Node head)
{
 // Your code here
   Node t1=head;
  Node t2=head.next;

   while(t2!=null&&t2.next!=null){
      t1.next=t1.next.next;
      t1=t1.next.[redacted];
      t2.next=t2.next.next;
      t2=t2.next[redacted];
   }

  Node temp=reverse(t2);
   t1.next=temp;
   while(head!=null){
      System.out.print(head.data+" ");
      head=head.next;
   }



}
}
```

```
static void unfold(Node head)
  {
    Node l1 = head;
    Node l2 = head.next;
    Node temp = l2;

    while(l2 != null && l2.next != null) {
      l1.next = l1.next.next;
      l2.next = l2.next.next;
      l1 = l1.next;
      l2 = l2.next;

    }

    Node l2Rev = reverse(temp);

    l1.next = l2Rev;

    printLL(head);

  }
```

6

1 6 2 5 3 4



1 2 3 4 5 6

Problem

al X

Sorted

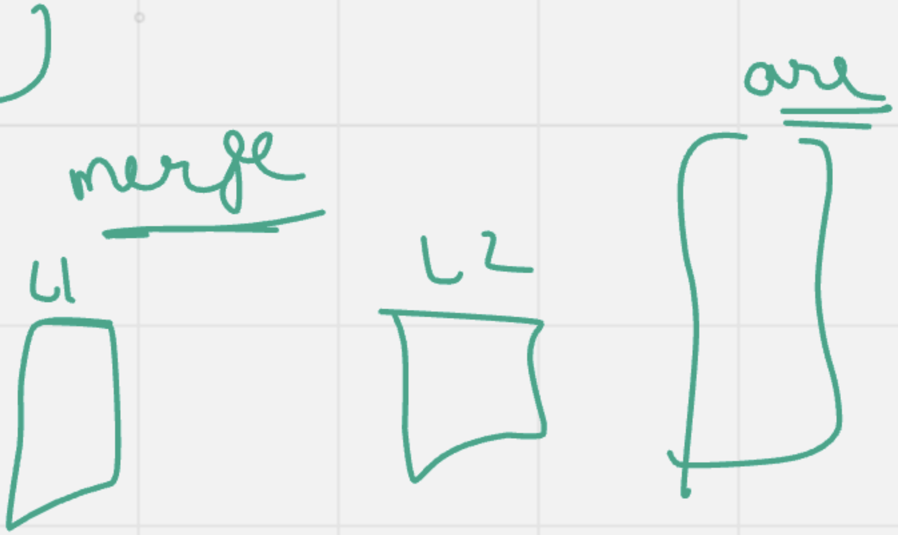
✓ L1 → 5 → 10 → 20 → null

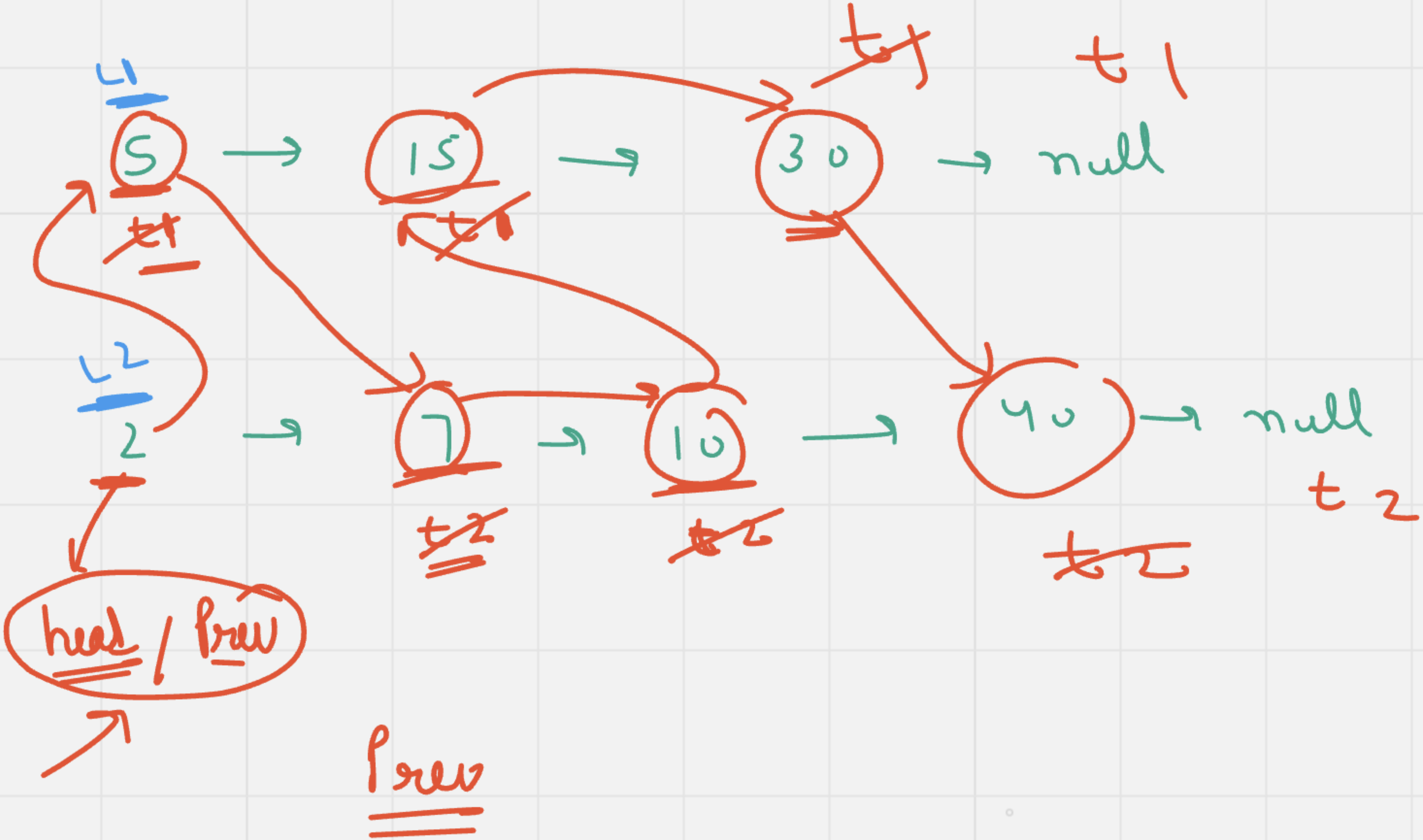
✓ L2 → 1 → 2 → 7 → 80 → null

merge these two into one → sorted

merge Sort

→ merge




L1
5
15
30
null
~~t1~~
~~t1~~
~~t1~~
t1
L2
2
7
10
40
null
~~t2~~
~~t2~~
~~t2~~
t2
head / Prev
Prev

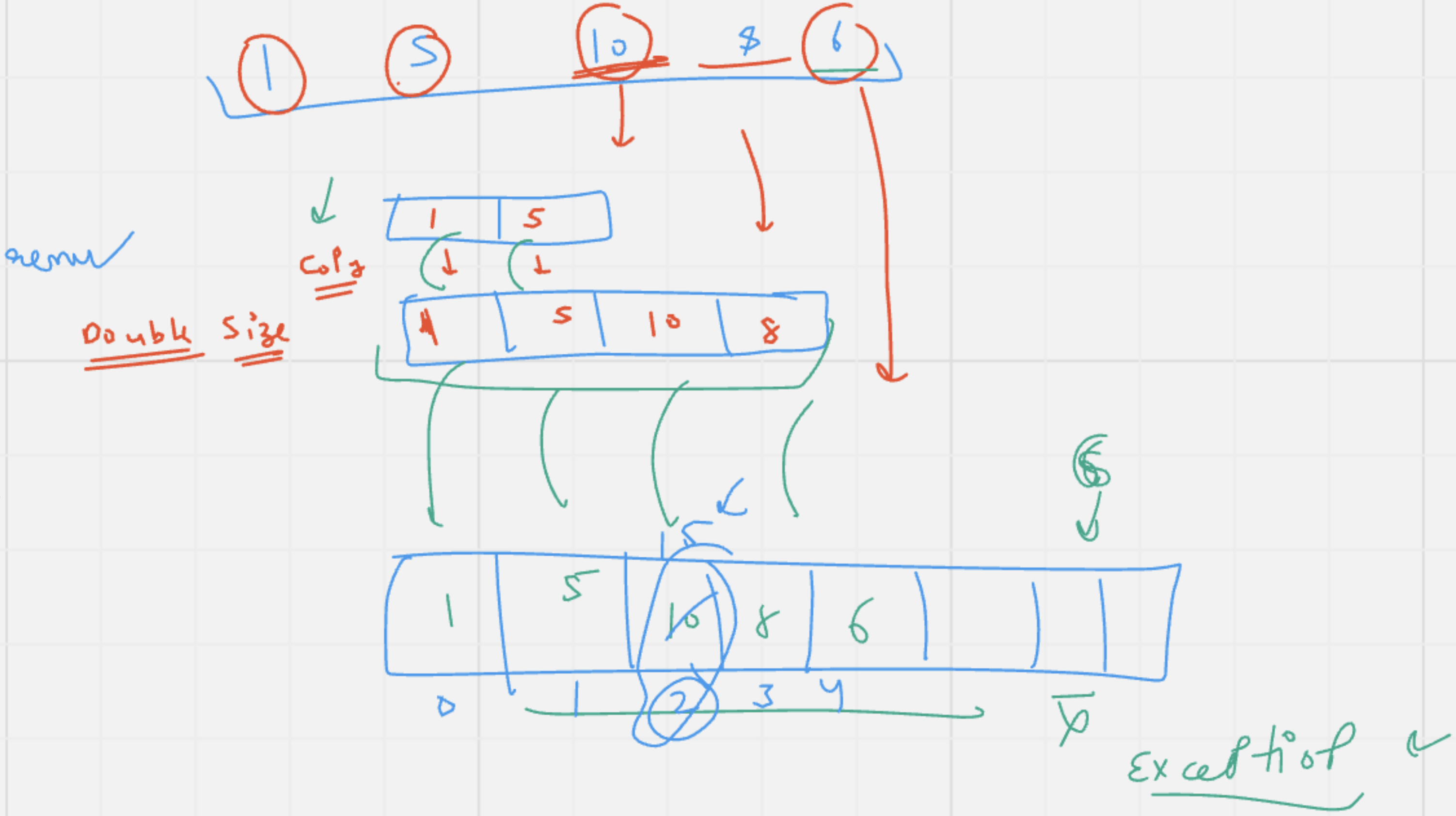
one topic OOPS

✓ Dynamic Array Implemen

Regarding arraylist

- add()
- get()
- size()
- remove()
- isEmpty
- set
- contains
- toString

remu


1
5
10
8
6
copy
1
5
Double Size
1
5
10
8
1
5
10
8
6
0
1
2
3
4
5
6
Exception