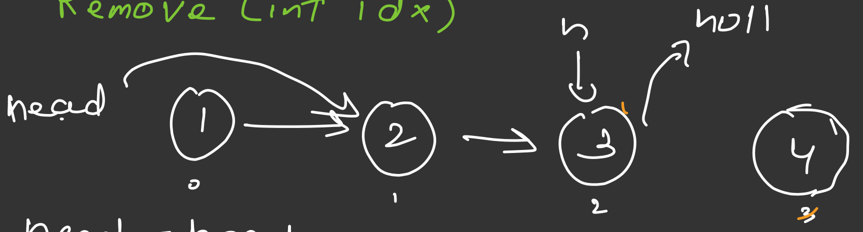
Remove (int idx)

idx = 3



head = head.next

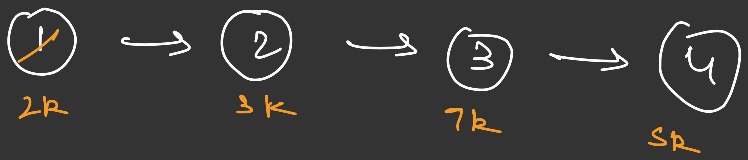
Node n = getNode(idx -1);

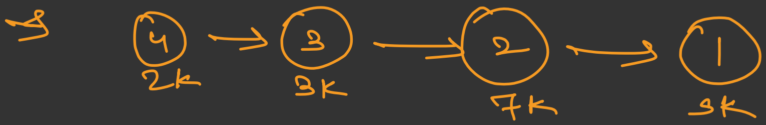
n.next = n.next.next;

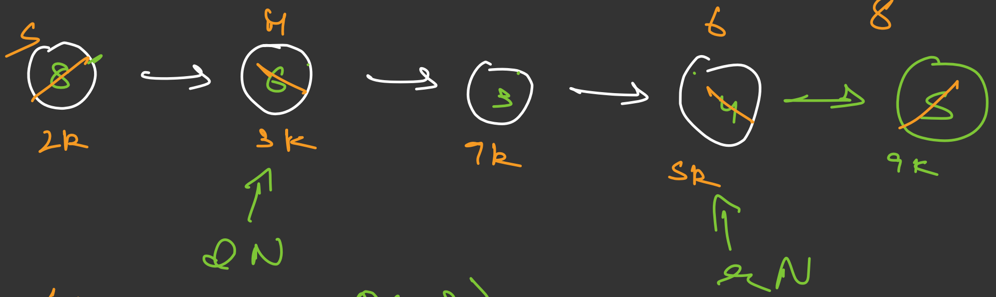
node(4)

t.next

Ques Reverse data iterative (ReverseD1)





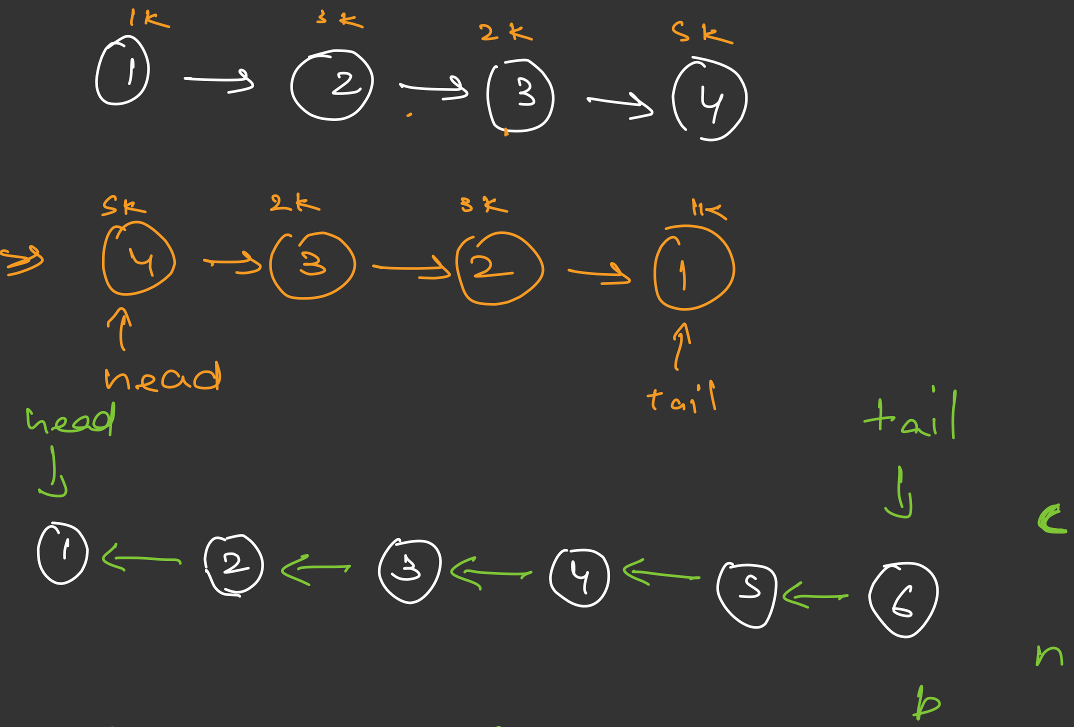


o = ~~0~~ 1

O(n²)

e = ~~4~~ 3

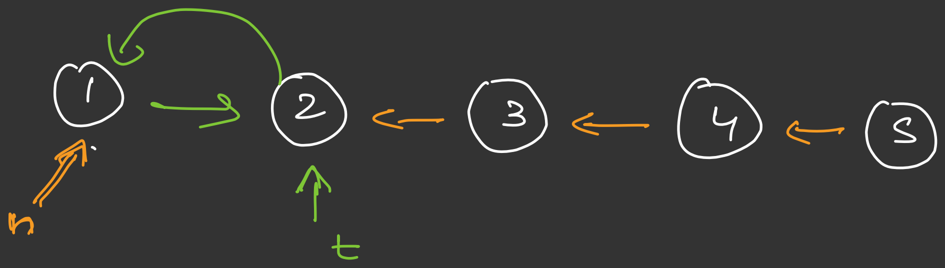
Ques
Reverse Pointer Iterative



```
p = null
n
       while ( c != null ) {
           n = c.next;
           c.next = p;
  ->       p = c;
           c = n;
       }
       head
```

```
Node t = h
h = tail
tail = t
```

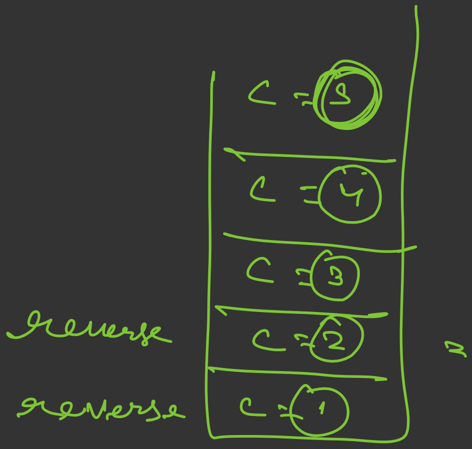
Ques Reverse Pointers Recursive



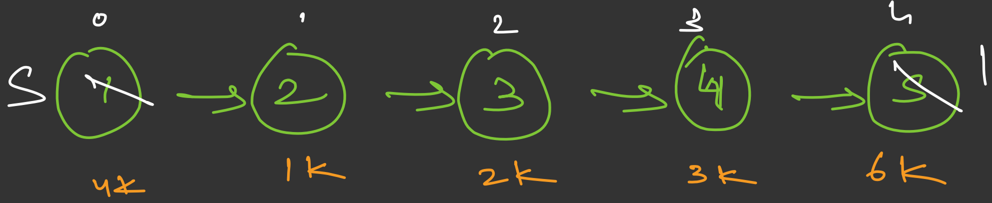
```
reverse ( Node     ) {

    → reverse(c.next);
      Node  h    = c.next;
    → h.next  =  c;
      c.next  =  null;

}
```

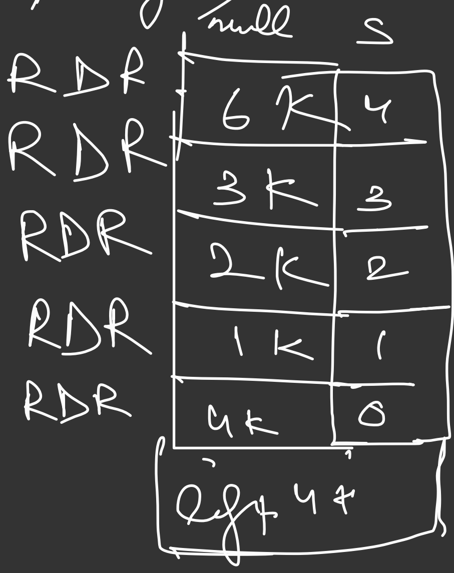


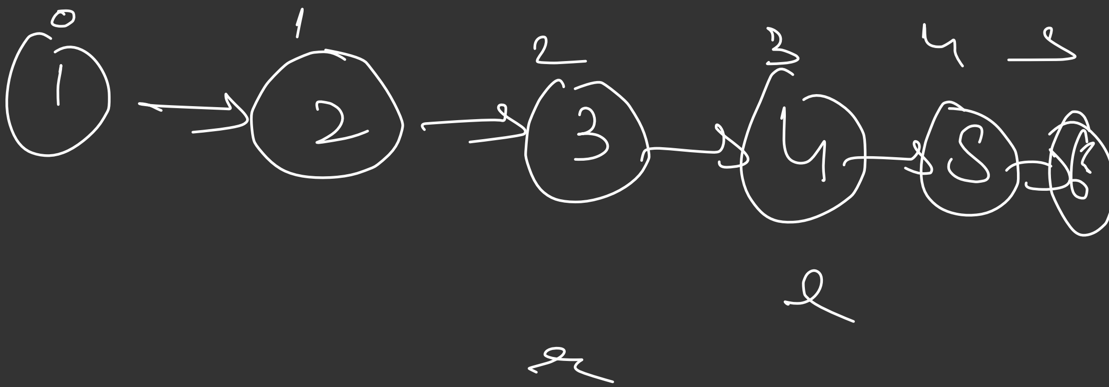
Reverse D R



```
left = head;
    Reverse DR (Node n, int i) {
    if(n = = null) return;

        Reverse DR(n.next, i+1)
        swap(left, right)
```





LL → ① → 3 pointer

② left right (Reverse LR)

4 types

int data

Node next