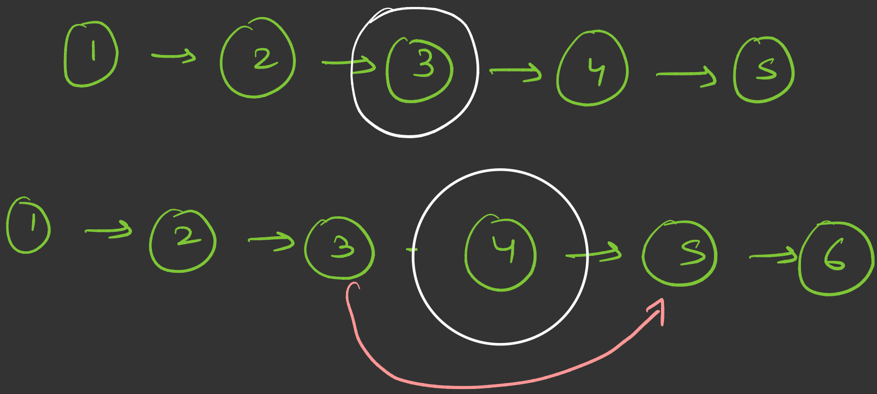
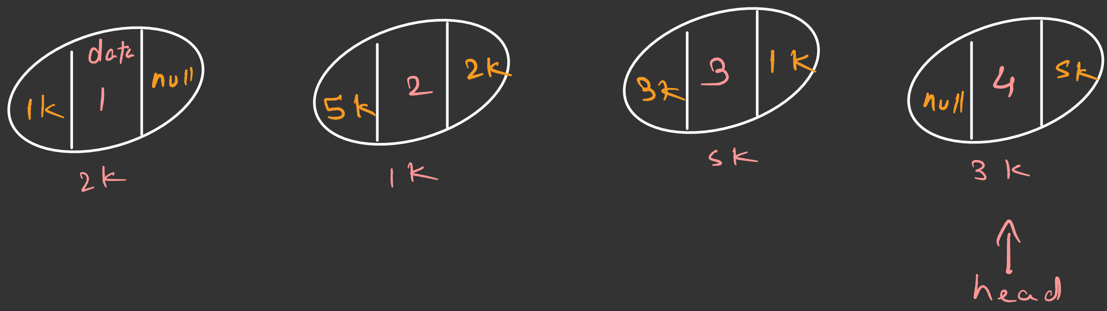
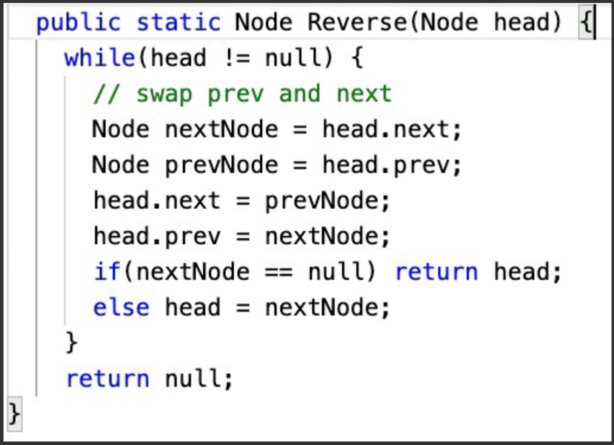
Ques Delete middle node



1. Calculate size
2. go to [size/2] node
3. Delete next node

Ques Reverse a Doubly LL





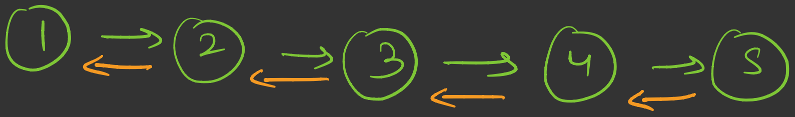
```java
public static Node Reverse(Node head) {
  while(head != null) {
    // swap prev and next
    Node nextNode = head.next;
    Node prevNode = head.prev;
    head.next = prevNode;
    head.prev = nextNode;
    if(nextNode == null) return head;
    else head = nextNode;
  }
  return null;
}
```

nn = null

pn = 5k

10:20

Ques DLL → delete kth from end



size = 5

k = 2

Ques 4 algorithms to reverse a singly LL

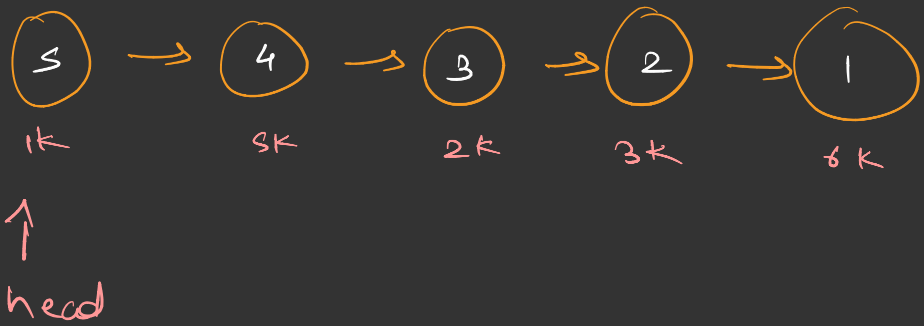
Reverse Data Iterative

Reverse Data Recursive

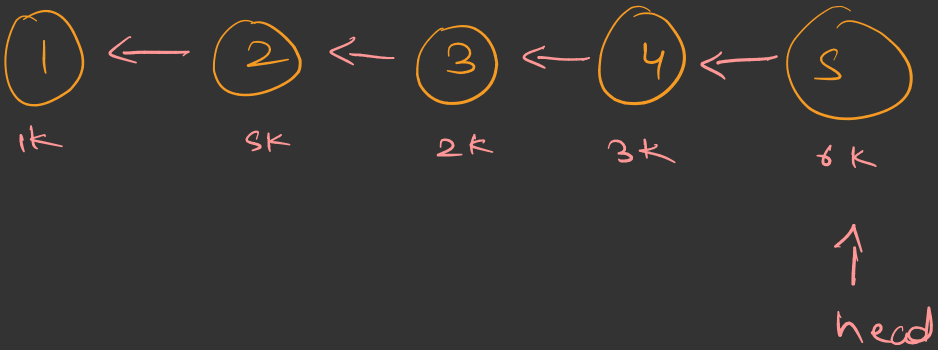
Reverse Pointer Iterative

Reverse Pointer Recursive

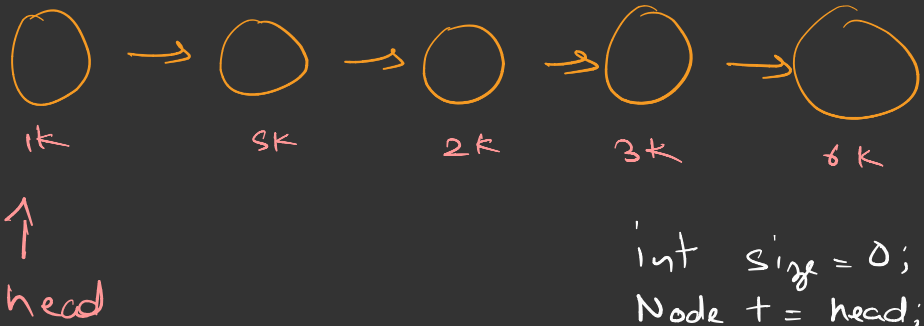
Data



Pointer



1 Reverse Data Iterative



```
int size = 0;
Node t = head;
 while (t != null) {
     t = t.next;
     size++;
 }
```

// Calculate Size

```
int l = 0, r = size - 1;
while (l < r) {
   Node ln = get(l); // O(n)
   Node rn = get(r);
   swap(ln, rn);
   l++; r--;
}
```

n

TC = O(n^2)