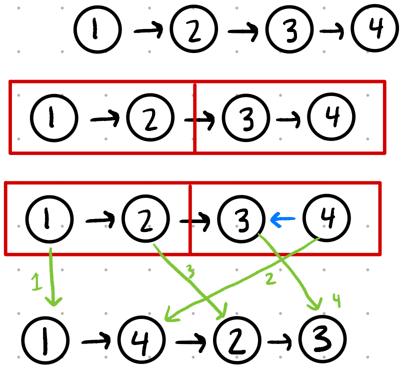
Reorder List

Ex 1



Split list into 2 parts, using Slow and Fast Pointers

Reverse Next Pointers of 2nd Part

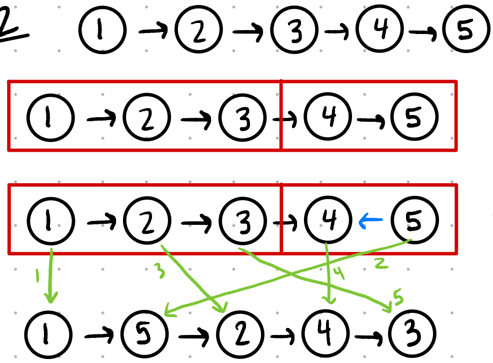
Grab head of Part 1, then head of Part 2. Increment in both Parts and repeat.

first Node stays same

insert behind Node 2

So start from 2nd Node and last Node

Ex 2

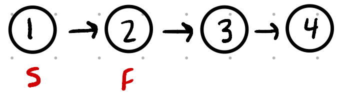


Split list into 2 parts, using Slow and Fast Pointers

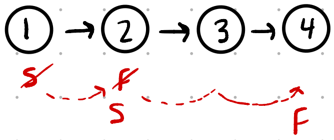
Reverse Next Pointers of 2nd Part

Grab head of Part 1, then head of Part 2. Increment in both Parts and repeat.

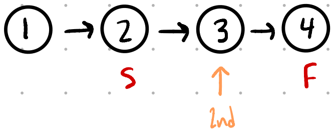
Let's Code it Now



S & F intially start here, then S increments by 1 and F increments by 2.

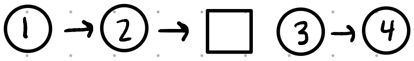


```
slow, fast = head, head.next
while fast and fast.next:
    slow = slow.next
    fast = fast.next.next
```

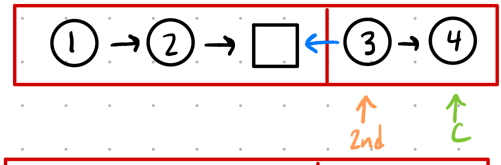


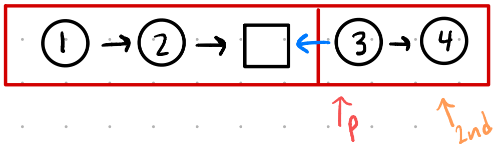
Now Mark 2nd Part of the List

```
second = slow.next
```

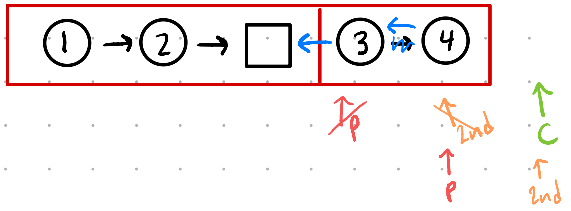


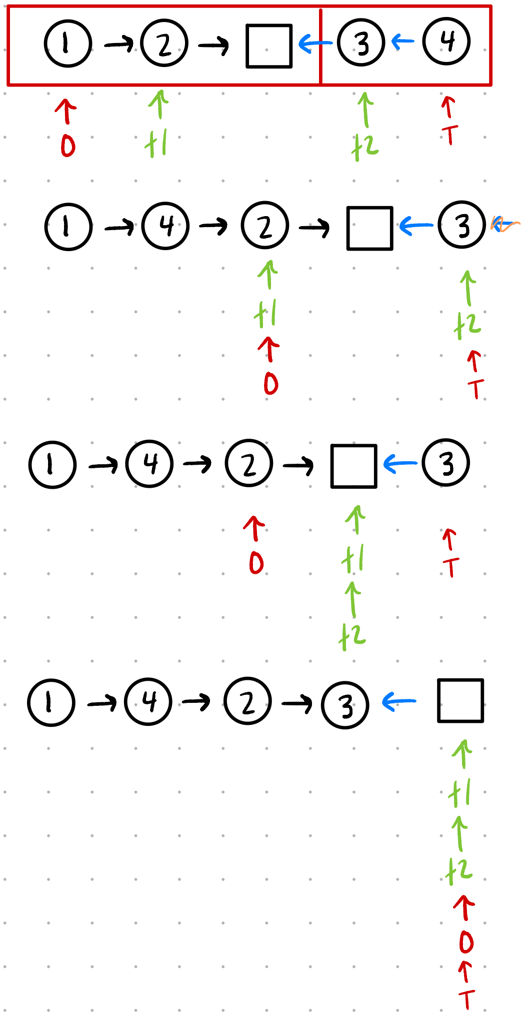
```
slow.next = None   # to seperate the Parts
prev = None
```





```
While second:
    Current = second.next
    second.next = prev
    prev = second
    second = Current
```





```
one, two = head, prev
while two :
    temp1, temp2 = one.next, two.next
    one.next = two
    two.next = temp1
    one, two = temp1, temp2
```