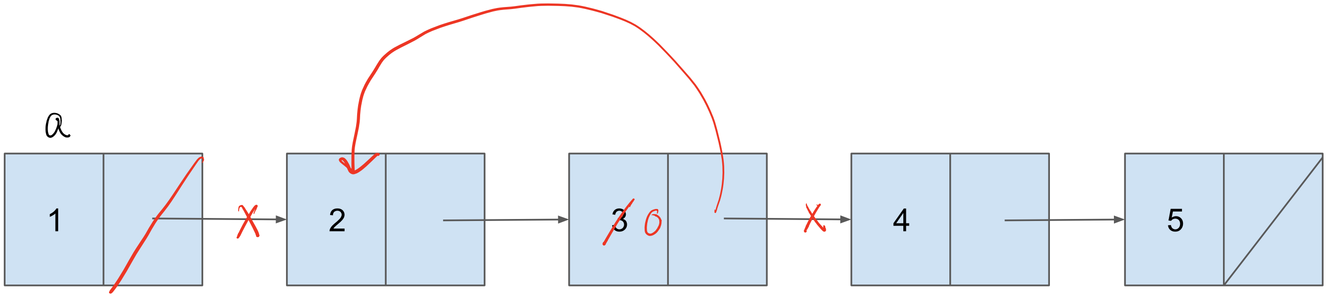
Linked Lists

Attendance Password: epic

Announcements

- Attendance on Gradescope
    - password: epic
- Homework 3 released tonight!
    - *Screenshots*: submit a screenshot of your code for each problem, along with an explanation of your approach
    - *Self Grades:* self grade your problems using the number of test cases passed on Leetcode
- Optional Big O / Time-Space Complexity review session
    - Monday, February 22nd 5-6 PM
- Instructor OH after lecture
    - 6:30 - 7:30 PM in this zoom meeting!

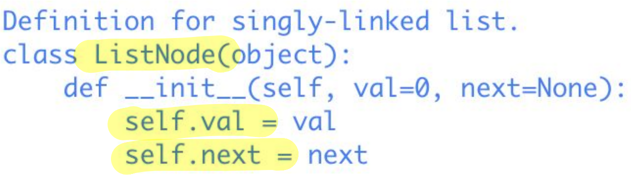
Linked Lists in Python



a.next.next.val = 0

a.next.next.next = a.next

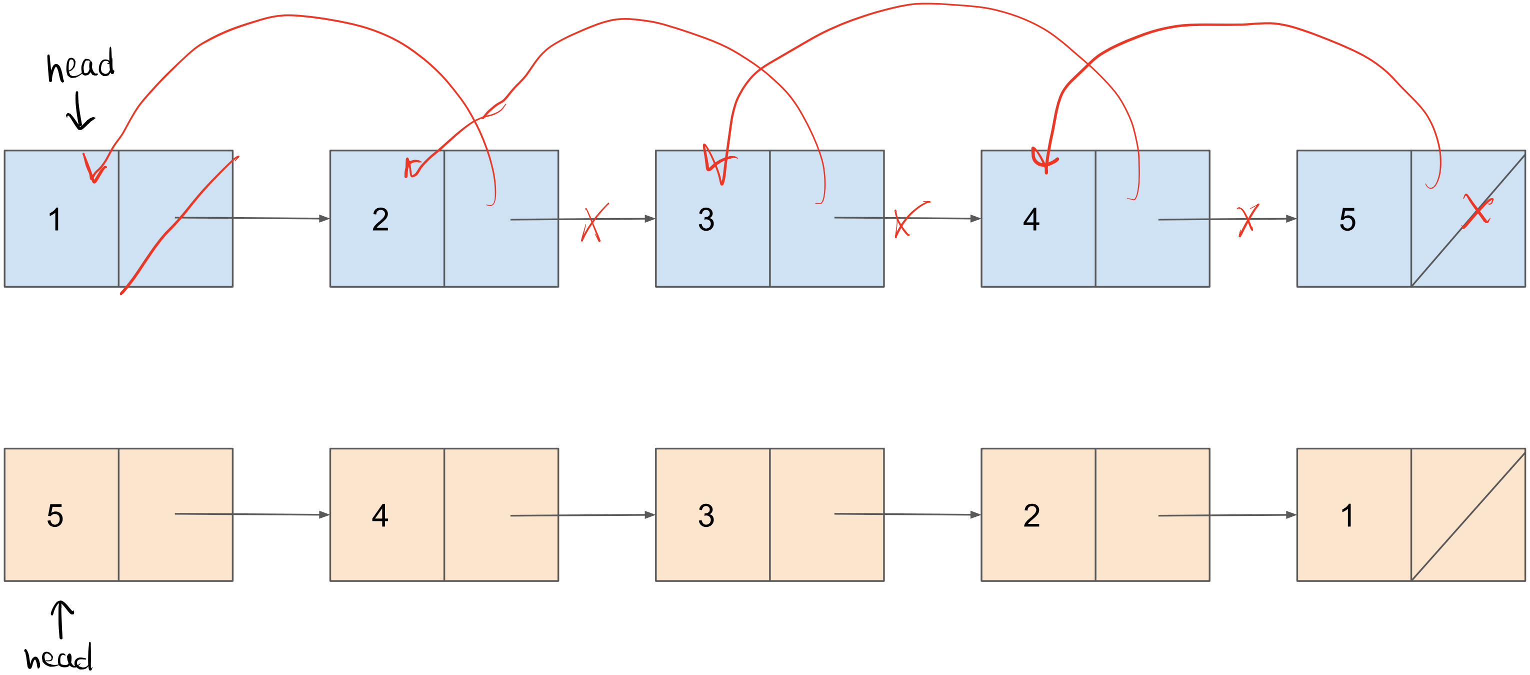
a.next = None



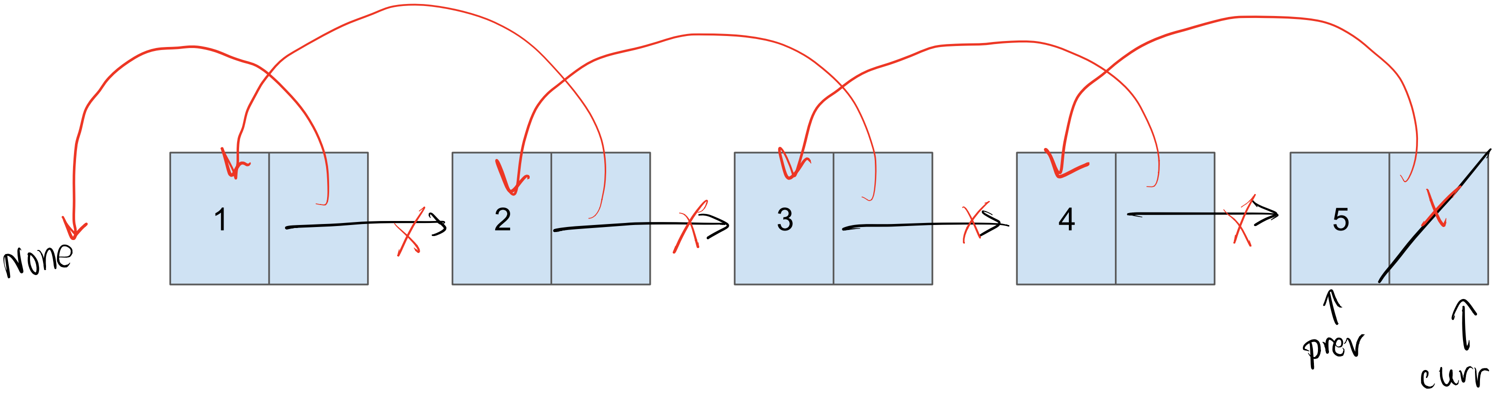
```
Definition for singly-linked list.
class ListNode(object):
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next
```

Reverse a linked list

Reversing a Linked List



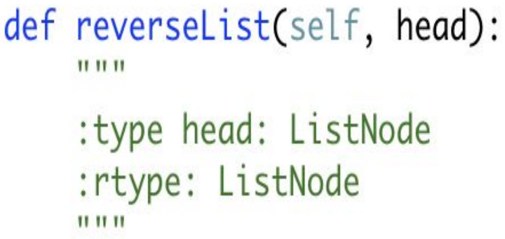
Iterative Solution. What do we need to keep track of?



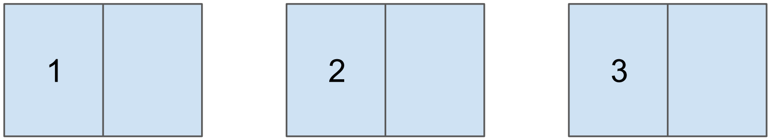
```
nxt = curr.next
curr.next = prev
prev = curr
curr = nxt
```

return prev

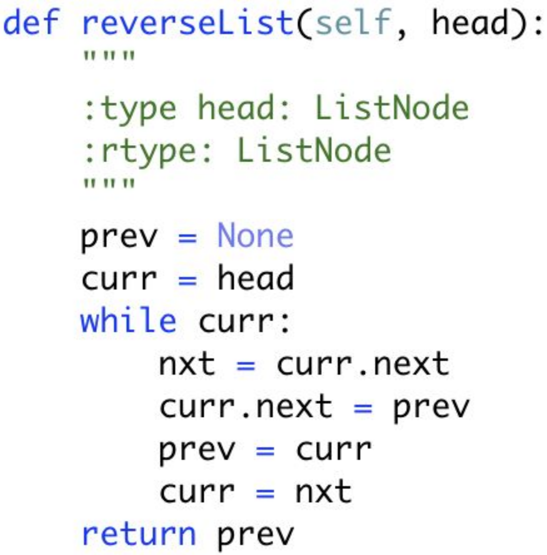
Iterative Solution.



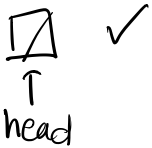
```
def reverseList(self, head):
    """
    :type head: ListNode
    :rtype: ListNode
    """
    curr =head
    prev = None
    while curr:
        nxt = curr.next
        curr.next=prev
        prev = curr
        curr = nxt
  return prev
```

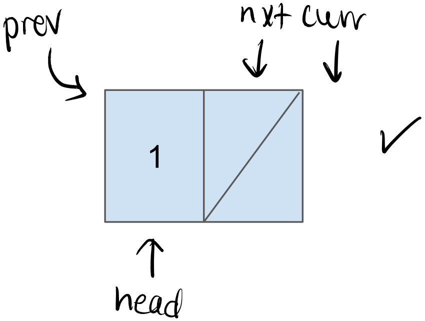


Iterative Solution. Edge Cases?



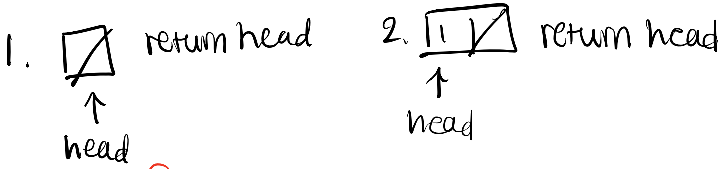
```
def reverseList(self, head):
    """
    :type head: ListNode
    :rtype: ListNode
    """
    prev = None
    curr = head
    while curr:
        nxt = curr.next
        curr.next = prev
        prev = curr
        curr = nxt
    return prev
```

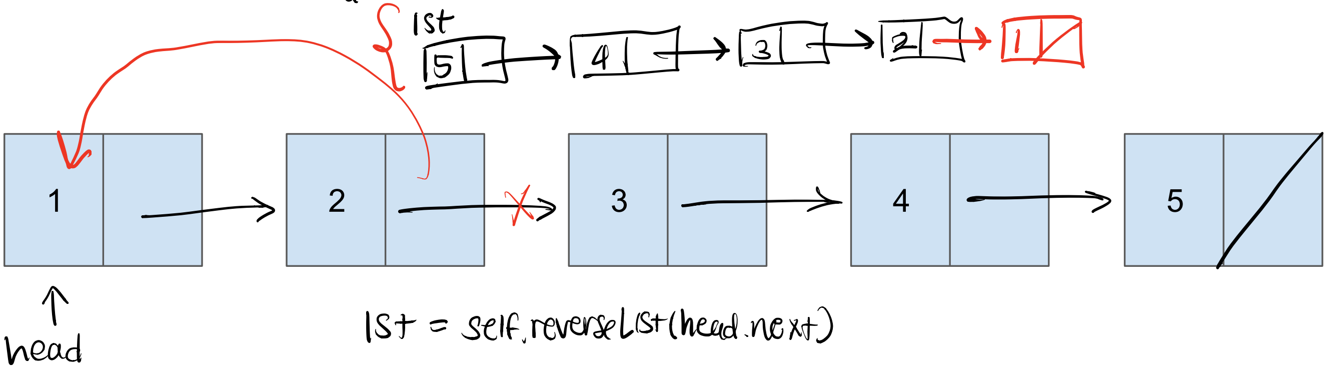




Recursive Solution. How do we break it into smaller subproblems?

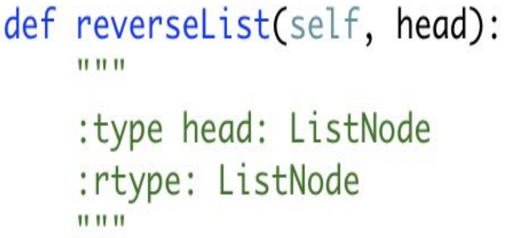
Base Cases:





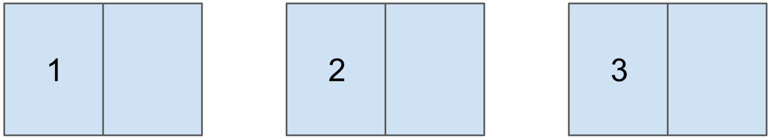
```
lst = self.reverseList(head.next)
head.next.next = head
head.next = None
return lst
```

Recursive Solution.

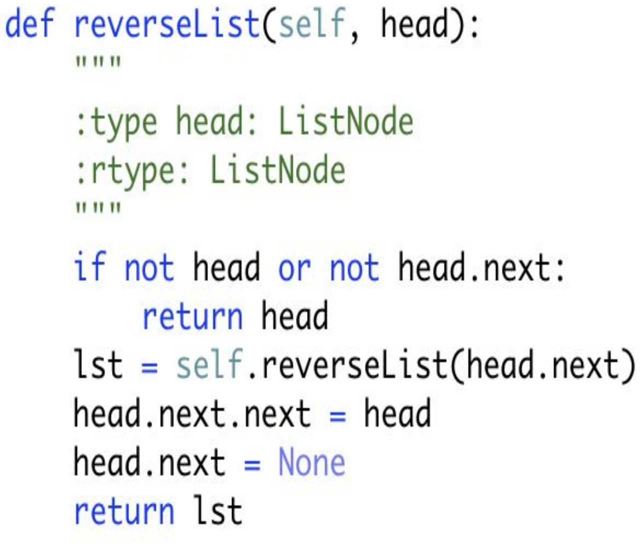


```
def reverseList(self, head):
    """
    :type head: ListNode
    :rtype: ListNode
    """
```

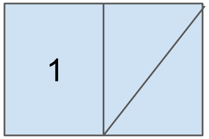
```
    if head is None or head.next is None:
        return head
    lst = self.reverseList(head.next)
    head.next.next = head
    head.next = None
    return lst
```



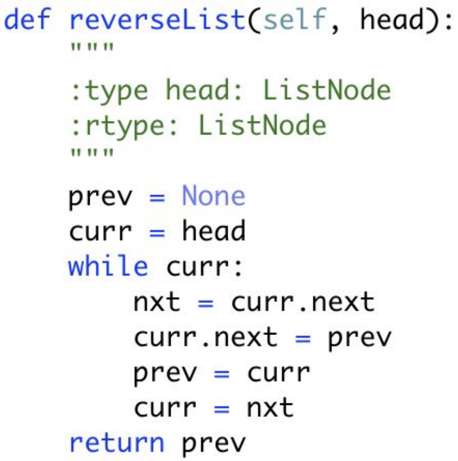
Recursive Solution. Edge Cases?



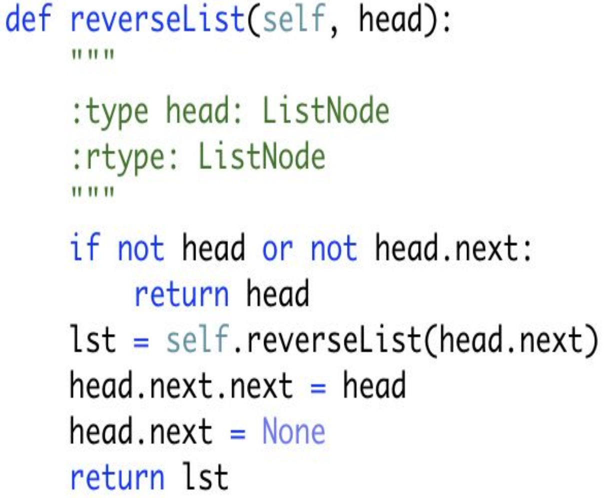
```python
def reverseList(self, head):
    """
    :type head: ListNode
    :rtype: ListNode
    """
    if not head or not head.next:
        return head
    lst = self.reverseList(head.next)
    head.next.next = head
    head.next = None
    return lst
```



Time/Space Complexity?



```python
def reverseList(self, head):
    """
    :type head: ListNode
    :rtype: ListNode
    """
    prev = None
    curr = head
    while curr:
        nxt = curr.next
        curr.next = prev
        prev = curr
        curr = nxt
    return prev
```



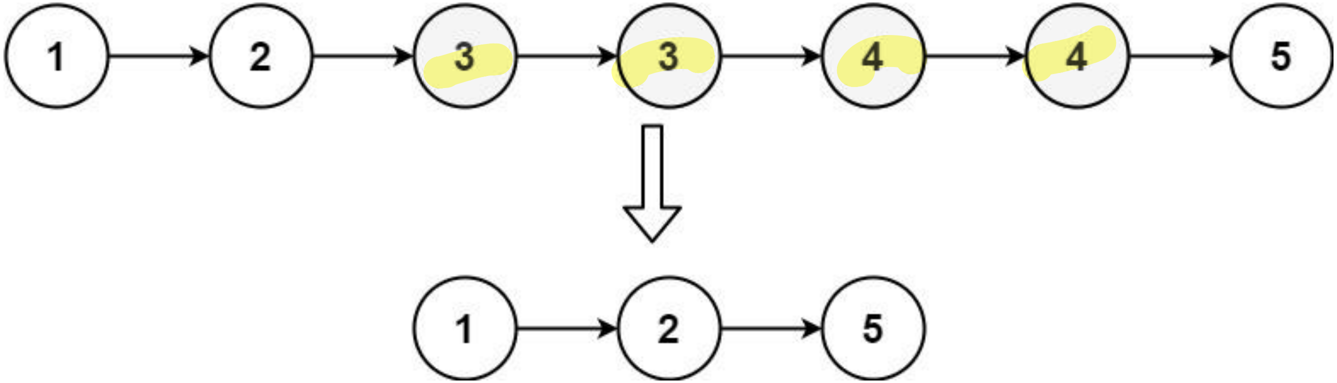
```python
def reverseList(self, head):
    """

    :type head: ListNode
    :rtype: ListNode
    """

    if not head or not head.next:
        return head
    lst = self.reverseList(head.next)
    head.next.next = head
    head.next = None
    return lst
```

Time: $O(n)$	Space: $O(1)$ recursive: $o(n)$

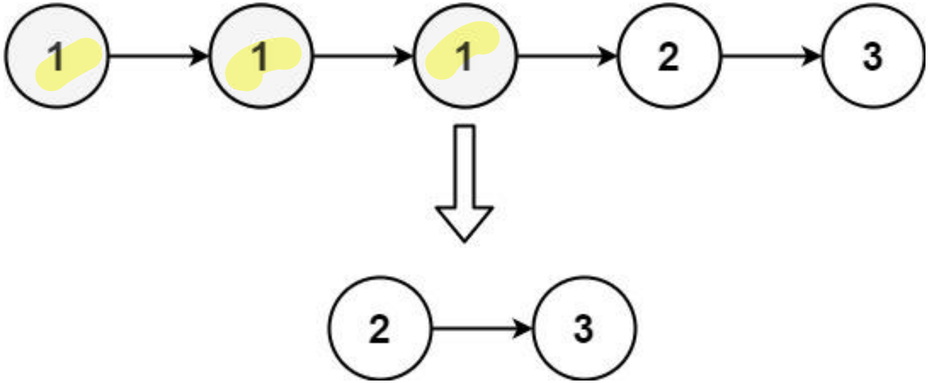
Removing duplicates from a sorted list (II)

The Problem

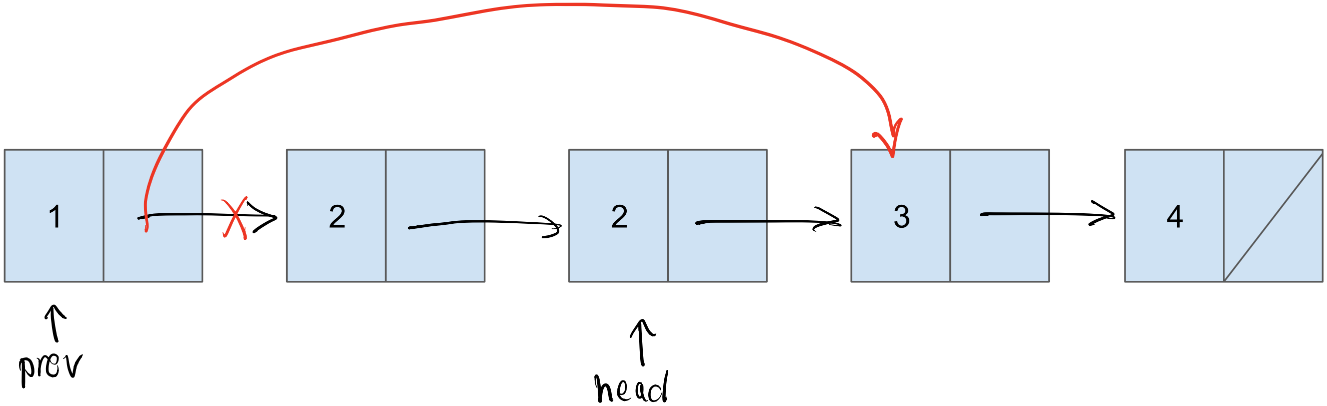
Given the head of a linked list, delete all nodes that have duplicate numbers, leaving only distinct numbers from the original list. Return the list sorted as well.



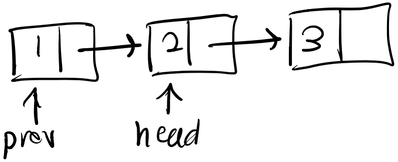
Another Example



The Approach

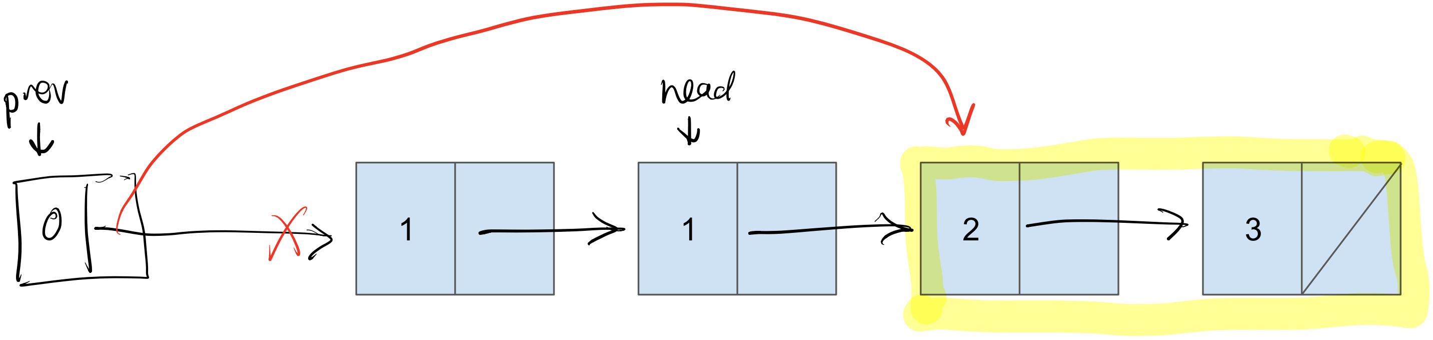


```
while head.next and head.val == head.next.val:
        head = head.next
prev.next = head.next
```



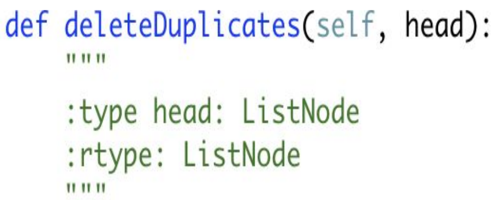
```
prev = prev.next
head = head.next
```

Edge Case: First node is removed?



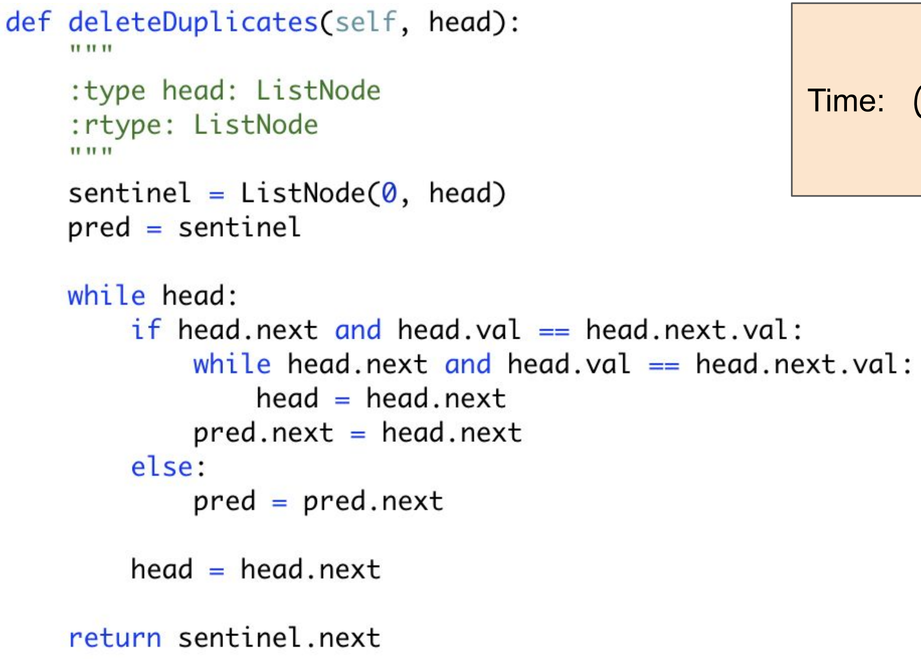
return prev.next

Solution



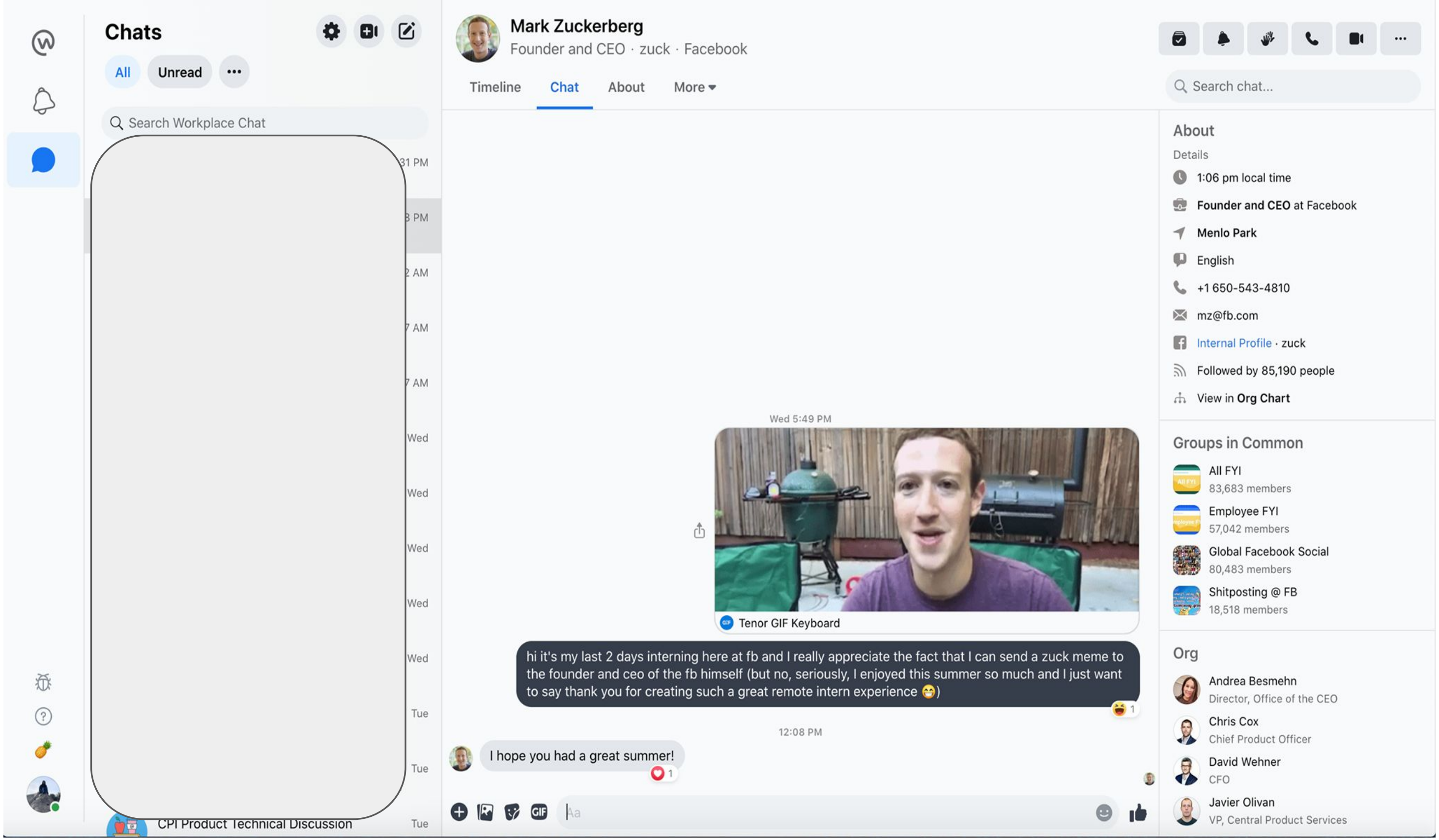
```
def deleteDuplicates(self, head):
    """
    :type head: ListNode
    :rtype: ListNode
    """
    sentinel = ListNode(0, head)
    prov = sentinel
    while head:
        if head.next and head.val == head.next.val:
            while head.next and head.val == head.next.val:
                head = head.next
            prov.next = head.next
        else:
            prov = prev.next
        head = head.next
    return sentinel.next
```

Time/Space Complexity?



```python
def deleteDuplicates(self, head):
    """
    :type head: ListNode
    :rtype: ListNode
    """
    sentinel = ListNode(0, head)
    pred = sentinel

    while head:
        if head.next and head.val == head.next.val:
            while head.next and head.val == head.next.val:
                head = head.next
            pred.next = head.next
        else:
            pred = pred.next

        head = head.next

    return sentinel.next
```

Time: O(n)	Space: O(1)



Chats

All Unread ···

Search Workplace Chat

[redacted]

31 PM

3 PM

2 AM

7 AM

7 AM

Wed

Wed

Wed

Wed

Wed

Tue

Tue

CPI Product Technical Discussion Tue

Mark Zuckerberg
Founder and CEO · zuck · Facebook

Timeline Chat About More ▾

Search chat...

Wed 5:49 PM

Tenor GIF Keyboard

hi it's my last 2 days interning here at fb and I really appreciate the fact that I can send a zuck meme to the founder and ceo of the fb himself (but no, seriously, I enjoyed this summer so much and I just want to say thank you for creating such a great remote intern experience 😁)

😝 1

12:08 PM

I hope you had a great summer!

❤️ 1

Aa

About

Details

1:06 pm local time

Founder and CEO at Facebook

Menlo Park

English

+1 650-543-4810

mz@fb.com

Internal Profile · zuck

Followed by 85,190 people

View in **Org Chart**

Groups in Common

All FYI	83,683 members
Employee FYI	57,042 members
Global Facebook Social	80,483 members
Shitposting @ FB	18,518 members

Org

Andrea Besmehn	Director, Office of the CEO
Chris Cox	Chief Product Officer
David Wehner	CFO
Javier Olivan	VP, Central Product Services

kinda irrelevant meme of the day

discussion!

1: this zoom room

2: https://berkeley.zoom.us/j/99755258402

3: https://berkeley.zoom.us/j/97505184820