

Reverse a Sub-list (medium)

We'll cover the following ^

- Problem Statement
- Try it yourself
- Solution
 - Code
 - Time complexity
 - Space complexity
- Similar Questions

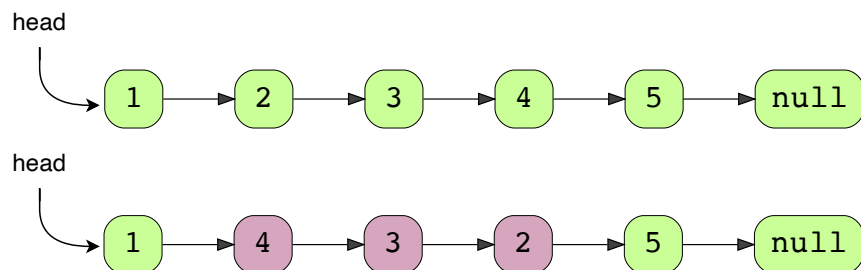
Problem Statement

Given the head of a LinkedList and two positions 'p' and 'q', reverse the LinkedList from position 'p' to 'q'.

Example:

Original List:

p=2, q=4



Try it yourself

Try solving this question here:

 Java





 Python3

 JS

 C++

```
1 from __future__ import print_function
2
3
4 class Node:
5     def __init__(self, value, next=None):
6         self.value = value
7         self.next = next
8
9     def print_list(self):
10        temp = self
11        while temp is not None:
12            print(temp.value, end=" ")
13            temp = temp.next
14        print()
15
```

```
16
17 def reverse_sub_list(head, p, q):
18     # TODO: Write your code here
19     return head
20
21
22 def main():
23     head = Node(1)
24     head.next = Node(2)
25     head.next.next = Node(3)
26     head.next.next.next = Node(4)
27     head.next.next.next.next = Node(5)
28
```



Solution

The problem follows the **In-place Reversal of a LinkedList** pattern. We can use a similar approach as discussed in Reverse a LinkedList

(<https://www.educative.io/collection/page/5668639101419520/5671464854355968/4519653420302336/>). Here are the steps we need to follow:





1. Skip the first $p-1$ nodes, to reach the node at position p .
2. Remember the node at position $p-1$ to be used later to connect with the reversed sub-list.
3. Next, reverse the nodes from p to q using the same approach discussed in Reverse a LinkedList

(<https://www.educative.io/collection/page/5668639101419520/5671464854355968/4519653420302336/>).


4. Connect the $p-1$ and $q+1$ nodes to the reversed sub-list.

Code

Here is what our algorithm will look like:

 Java	 Python3	 C++	 JS
--	---	---	--

```
1 from __future__ import print_function
2
3
4 class Node:
5     def __init__(self, value, next=None):
6         self.value = value
7         self.next = next
8
9     def print_list(self):
10         temp = self
11         while temp is not None:
12             print(temp.value, end=" ")
13             temp = temp.next
14         print()
15
16
17 def reverse_sub_list(head, p, q):
18     if p == q:
19         return head
20
21     # after skipping 'p-1' nodes, current will point to 'p'th node
22     current, previous = head, None
23     i = 0
```



```

24 while current is not None and i < p - 1:
25     previous = current
26     current = current.next
27     i += 1
28

```



Time complexity

The time complexity of our algorithm will be $O(N)$ where 'N' is the total number of nodes in the LinkedList.

Space complexity

We only used constant space, therefore, the space complexity of our algorithm is $O(1)$.

Similar Questions

Problem 1: Reverse the first 'k' elements of a given LinkedList.

Solution: This problem can be easily converted to our parent problem; to reverse the first 'k' nodes of the list, we need to pass $p=1$ and $q=k$.

Problem 2: Given a LinkedList with 'n' nodes, reverse it based on its size in the following way:

1. If 'n' is even, reverse the list in a group of $n/2$ nodes.
2. If n is odd, keep the middle node as it is, reverse the first ' $n/2$ ' nodes and reverse the last ' $n/2$ ' nodes.

Solution: When 'n' is even we can perform the following steps:

1. Reverse first ' $n/2$ ' nodes: `head = reverse(head, 1, n/2)`
2. Reverse last ' $n/2$ ' nodes: `head = reverse(head, n/2 + 1, n)`

When 'n' is odd, our algorithm will look like:

1. `head = reverse(head, 1, n/2)`
2. `head = reverse(head, n/2 + 2, n)`

Please note the function call in the second step. We're skipping two elements as we will be skipping the middle element.

← Back

Next →

Reverse a LinkedList (easy)

Reverse every K-element Sub-list (me...

✓ Mark as Completed

🔔 Report an Issue

🔗 Ask a Question

(https://discuss.educative.io/tag/reverse-a-sub-list-medium__pattern-in-place-reversal-of-a-linkedlist__grokking-the-coding-interview-patterns-for-coding-questions)

