

Introduction

(medium)

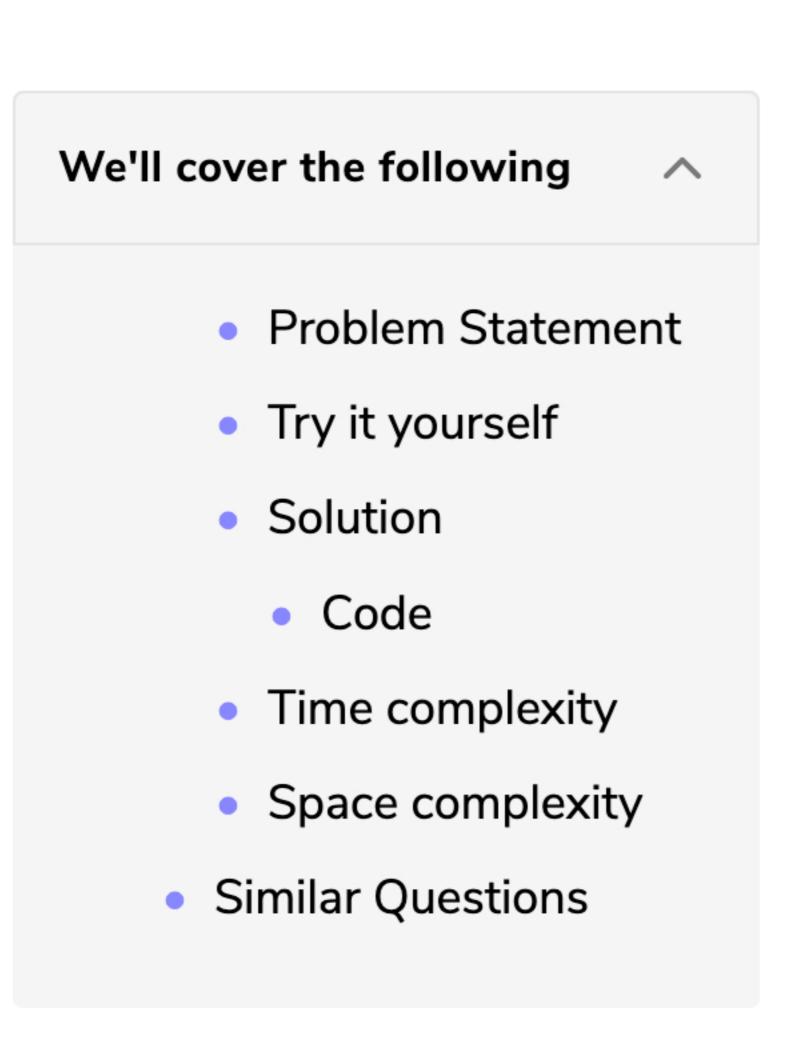
Reverse a LinkedList (easy)

Reverse a Sub-list (medium)

Reverse every K-element Sub-list

educative

Reverse a Sub-list (medium)



Problem Statement

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

€

? Ask a Question

```
Example:
                          head
   Original List:
   p=2, q=4
                          head
```

Try it yourself

```
Try solving this question here:
             Python3
                           JS JS
                                      G C++
  👙 Java
         ListNode(int value) {
    8
           this.value = value;
    9
   10
       class ReverseSubList {
   13
         public static ListNode reverse(ListNode head, int p, int q) {
   15
           // TODO: Write your code here
   16
           return head;
   18
   19
         public static void main(String[] args) {
   20
           ListNode head = new ListNode(1);
           head.next = new ListNode(2);
   21
   22
           head.next.next = new ListNode(3);
           head.next.next.next = new ListNode(4);
   23
   24
           head.next.next.next.next = new ListNode(5);
   25
           ListNode result = ReverseSubList.reverse(head, 2, 4);
   26
           System.out.print("Nodes of the reversed LinkedList are: ");
           while (result != null) {
   28
   29
             System.out.print(result.value + " ");
   30
             result = result.next;
   31
   32
   Run
                                                                             Save
                                                                                      Reset
```

The problem follows the In-place Reversal of a LinkedList pattern. We can use a similar

Solution

approach as discussed in Reverse a LinkedList. 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. 4. Connect the p−1 and q+1 nodes to the reversed sub-list.
- Code

Here is what our algorithm will look like:

```
Python3
                          ⓒ C++
                                      JS JS
  Java
       import java.util.*;
       class ListNode {
         int value = 0;
         ListNode next;
    6
         ListNode(int value) {
           this.value = value;
    8
    9
   10
       class ReverseSubList {
   13
         public static ListNode reverse(ListNode head, int p, int q) {
   14
           if (p == q)
   15
   16
             return head;
           // after skipping 'p-1' nodes, current will point to 'p'th node
           ListNode current = head, previous = null;
   19
           for (int i = 0; current != null && i < p - 1; ++i) {
   20
             previous = current;
             current = current.next;
   23
   24
           // we are interested in three parts of the LinkedList, part before index 'p', part bet
   25
           // 'q', and the part after index 'q'
   26
           ListNode lastNodeOfFirstPart = previous; // points to the node at index 'p-1'
           // after reversing the LinkedList 'current' will become the last node of the sub-list
   Run
                                                                                     Reset
                                                                            Save
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

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:

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.

1. If 'n' is even, reverse the list in a group of n/2 nodes.

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

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)

instead of the other way around. See how ①

skipping the middle element.

Interviewing soon? We've partnered with Hired so that companies apply to you,

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



Report an Issue