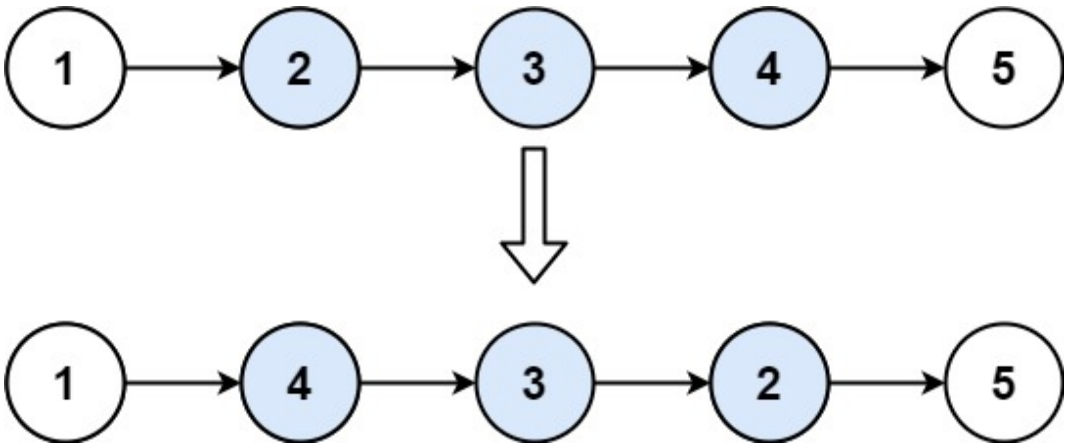
92. Reverse Linked List II

Created	@October 20, 2021 11:34 PM
Difficulty	Medium
LC Url	https://leetcode.com/problems/reverse-linked-list-ii/
Importance	
Tag	Array&Sorting
Video	

Given the `head` of a singly linked list and two integers `left` and `right` where `left <= right`, reverse the nodes of the list from position `left` to position `right`, and return *the reversed list*.

Example 1:




```
Input: head = [1,2,3,4,5], left = 2, right = 4
Output: [1,4,3,2,5]
```

Example 2:

```
Input: head = [5], left = 1, right = 1
Output: [5]
```

Constraints:

- The number of nodes in the list is `n`.

- `1 <= n <= 500`
- `500 <= Node.val <= 500`
- `1 <= left <= right <= n`

Follow up:

Could you do it in one pass?

Solution

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def reverseBetween(self, head: Optional[ListNode], left: int, right: int) -> Optional[ListNode]:
        def reverseN(head, right):
            if right == 1:
                self.successor = head.next
                return head
            last = reverseN(head.next, right-1)
            head.next.next = head
            head.next = self.successor
            return last

        if left == 1:
            return reverseN(head, right)

        head.next = self.reverseBetween(head.next, left-1, right-1)
        return head
```

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def reverseBetween(self, head: ListNode, left: int, right: int) -> ListNode:
        """
        Example:
        [1 -> 2 -> 3 -> 4 -> 5]
        """

        # edge case: left and right point the same
        if left == right:
            return head

        """
        current = 1
```

```
        previous = None
        """
        current = head
        previous = None

        # skip to left - 1 nodes
        for i in range(left - 1):
            previous = current
            current = current.next

        # the node before sublist
        """
        current = 2
        previous = 1
        lastNodeOfFirstPart = 1
        """
        lastNodeOfFirstPart = previous

        # after reversing; last node of sublist will be current
        """
        lastNodeOfSubList = 2
        """
        lastNodeOfSubList = current
        next = None

        # reverse until right
        for i in range(right - left + 1):
            next = current.next
            current.next = previous
            previous = current
            current = next

        """
        After the reverse
        1 -> <- 2 <- 3 <- 4

        current = 5
        """

        # connect the first part
        """
        1 -> 4 -> 3 -> 2
        """
        if lastNodeOfFirstPart:
            lastNodeOfFirstPart.next = previous
        else:
            head = previous

        # connect to the last part
        """
        lastNodeOfSubList = 2
        current = 5

        1 -> 4 -> 3 -> 2 -> 5
        """
        lastNodeOfSubList.next = current

        return head
```