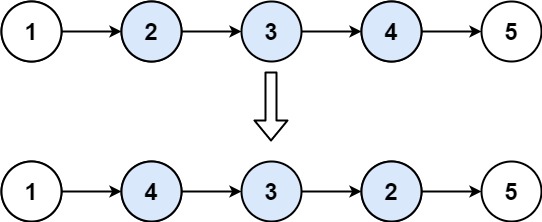
Given the head of a singly linked list and two integers left and right where left <= right, reverse the nodes of the list from position left to position right, and return *the reversed list*.

**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]

**Sol:**

/\*\*

\* Definition for singly-linked list.

\* public class ListNode {

\* int val;

\* ListNode next;

\* ListNode() {}

\* ListNode(int val) { this.val = val; }

\* ListNode(int val, ListNode next) { this.val = val; this.next = next; }

\* }

\*/

class Solution {

public ListNode reverseBetween(ListNode head, int left, int right) {

ListNode node = head;

int len = 0;

while(node != null){

len++;

node = node.next;

}

if(left == 1 && right == 1 && len == 1)

return head;

else if(left == right)

return head;

ListNode first = head, prev1 = null, sec = head;

left--; right--;

while(left-- >0){

prev1 = first;

first = first.next;

}

while(right-- >0){

sec = sec.next;

}

ListNode next;

ListNode curr = first, last = sec.next, prev = prev1;

while(curr != last){

next = curr.next;

curr.next = prev;

prev = curr;

curr = next;

}

if(prev1!= null)

prev1.next = prev;

else

head = prev;

first.next = last;

return head;

}

}