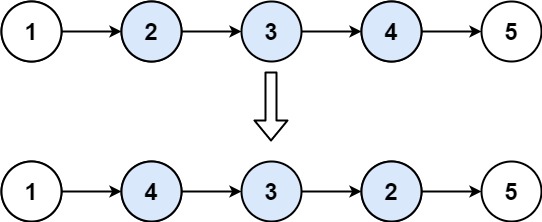
**92. Reverse Linked List II: -**

Medium Accepted: 703.2K Submissions: 1.5M Acceptance Rate: 46.3%

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?

**Code: -**

/\*\*

 \* Definition for singly-linked list.

 \* struct ListNode {

 \*     int val;

 \*     ListNode \*next;

 \*     ListNode() : val(0), next(nullptr) {}

 \*     ListNode(int x) : val(x), next(nullptr) {}

 \*     ListNode(int x, ListNode \*next) : val(x), next(next) {}

 \* };

 \*/

class Solution {

public:

    ListNode\* reverseBetween(ListNode\* head, int left, int right) {

        if(left == right)

            return head;

        ListNode \*prelow=NULL, \*low=NULL, \*high=NULL, \*posthigh=NULL;

        ListNode \*ptr = head, \*before = NULL, \*after = NULL;

        int ind = 1;

        while(ptr){

            if(ind == left-1)

                prelow = ptr;

            else if(ind == right+1)

                posthigh = ptr;

            if(left<=ind and ind<=right){

                if(ind == left)     low = ptr;

                if(ind == right)    high = ptr;

                after = ptr;

                ptr = ptr->next;

                after->next = before;

                before = after;

            }

            if(ind<left or ind>right)

                ptr = ptr->next;

            ++ind;

        }

        if(left == 1)

            head = high;

        if(prelow)

            prelow->next = high;

        if(posthigh)

            low->next = posthigh;

        return head;

    }

};

**T.C: - O(N)**

**S.C: - O(1)**