

[Description](#)
[Solution](#)
[Discuss \(999+\)](#)
[Submissions](#)

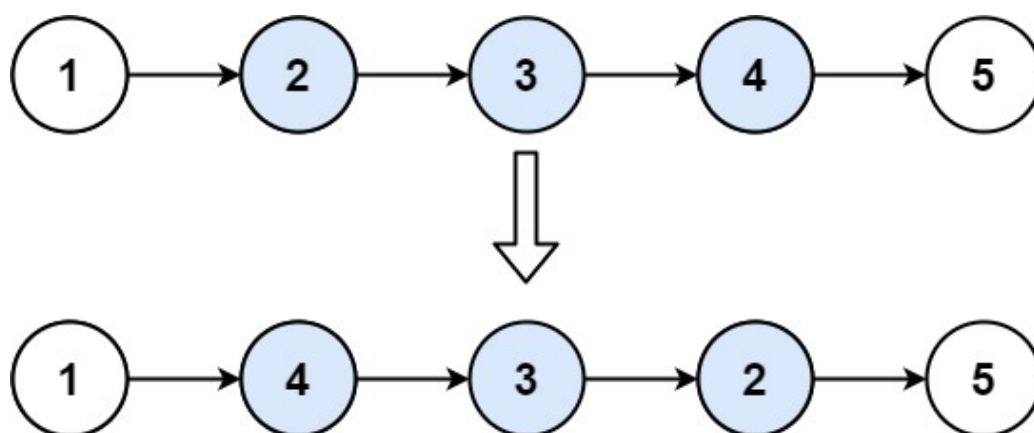
i C++

92. Reverse Linked List II

Medium
[7092](#)
[317](#)
[Add to List](#)
[Share](#)

Given the `head` of a singly linked list and two integers `left` and `right` where $left \leq right$, 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`.
- $1 \leq n \leq 500$
- $-500 \leq \text{Node.val} \leq 500$
- $1 \leq left \leq right \leq n$

Follow up: Could you do it in one pass?

```

1  /**
2   * Definition of singly-linked list.
3   * struct ListNode {
4   *     int val;
5   *     ListNode *next;
6   *     ListNode() : val(0), next(nullptr) {}
7   *     ListNode(int x) : val(x), next(nullptr) {}
8   *     ListNode(int x, ListNode *next) : val(x), next(next) {}
9   * };
10  */
11  class Solution {
12  public:
13      ListNode* reverseBetween(ListNode* head, int left, int right) {
14          if (left == right) return head;
15
16          // Find the node at position left-1
17          ListNode* prev = head;
18          for (int i = 1; i < left; i++) {
19              prev = prev->next;
20          }
21
22          // Reverse the list from left to right
23          ListNode* curr = prev->next;
24          prev->next = nullptr;
25          while (curr) {
26              curr->next = prev->next;
27              prev->next = curr;
28              curr = curr->next;
29          }
30          return head;
31      }
32  };
33  
```

Your previous code was

Console

[Problems](#)

[Pick One](#)

[Prev](#)

92/2344

[Next](#)

[Run Code](#)

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.