92. Reverse Linked List II ★

Question | Editorial Solution | My Submissions (/problems/reverse-linked-list-ii/submissions/)

Total Accepted: **83734** Total Submissions: **287671** Difficulty: **Medium**

Reverse a linked list from position *m* to *n*. Do it in-place and in one-pass.

For example:

Given `1->2->3->4->5->NULL` , *m* = 2 and *n* = 4,

return `1->4->3->2->5->NULL` .

Note:

Given *m*, *n* satisfy the following condition:

1 ≤ *m* ≤ *n* ≤ length of list.

Subscribe (/subscribe/) to see which companies asked this question

Show Tags

Show Similar Problems

Have you met this question in a real interview? Yes No

Discuss (https://leetcode.com/discuss/questions/oj/reverse-linked-list-ii)

Pick One (/problems/random-one-question/)

C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* reverseBetween(ListNode* head, int m, int n) {
        if (m < 1 || m > n) return NULL;

        vector<ListNode*> cache;
        ListNode* curNode = head; // 1
        ListNode* prev = NULL, *next = NULL;
        int index = 1;
        while (index <= n && curNode != NULL) {
            if (index >= m) {
                cache.push_back(curNode);
```

```
21              }
22              else if (m != 1 && index == m - 1) {
23                  prev = curNode;
24              }
25              curNode = curNode->next;
26              ++index;
27          }
28          next = curNode;
29
30          if (cache.size() != (n - m + 1)) {
31              return NULL;
32          }
33          else {
34              if (prev)
35                  prev->next = cache.back();
36              else
37                  head = cache.back();
38              for (int i = cache.size() - 1; i > 0; --i) {
39                  cache[i]->next = cache[i-1];
40              }
41              cache[0]->next = next;
42          }
43          return head;
44      }
45  };
```

Notes

Custom Testcase

Run Code | Submit Solution