

61. [rotate a list](#)

```
class Solution {
public:
    ListNode* rotateRight(ListNode* head, int k) {
        if (!head || !head->next || k == 0) return head;

        ListNode* temp = head;
        int length = 1;
        while (temp->next) {
            temp = temp->next;
            length++;
        }

        k = k % length;
        if (k == 0) return head;

        ListNode* newTail = head;
        for (int i = 1; i < length - k; ++i) {
            newTail = newTail->next;
        }

        ListNode* newHead = newTail->next;

        temp->next = head;

        newTail->next = nullptr;
    }
};
```

```
return newHead;

}

};
```

leetcode.com/problems/rotate-list/submissions/1542063812/

Problem List < > > >

Description Accepted x Editorial Solutions Submissions

All Submissions

Accepted 232 / 232 testcases passed
aashima_narula submitted at Feb 13, 2025 23:22

Editorial Solution

Runtime 0 ms | Beats 100.00%
Memory 16.34 MB | Beats 65.00%

100% 50% 0%

1ms 2ms 3ms 4ms

Code C++

```
/**
 * 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* rotateRight(ListNode* head, int k) {
        if (!head || !head->next || k == 0) return head;

        ListNode* temp = head;
        int length = 1;
        while (temp->next) {
            temp = temp->next;
            length++;
        }
        temp->next = head;
        int newHead = temp->next;
        for (int i = 0; i < k; i++) {
            newHead = newHead->next;
        }
        return newHead;
    }
};
```

Testcase Test Result

Case 1 Case 2 +

head =

[1,2,3,4,5]

k =

2

Source

leetcode.com/problems/reverse-linked-list-ii/submissions/1542056447/

Problem List

Run

Submit

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted 44 / 44 testcases passed

azohima_narula submitted at Feb 13, 2025 23:17

Editorial

Solution

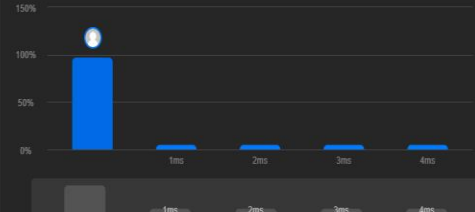
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

11.26 MB | Beats 38.74%



Code

C++

Auto

Ln 27, Col 1

```
1 /**
2  * Definition for 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     vector<int> reverseBetween(ListNode* head, int left, int right) {
14         if (!head) return {};
15         vector<int> ans;
16         while (head) {
17             ans.push_back(head->val);
18             head = head->next;
19         }
20         reverse(ans.begin() + left - 1, ans.begin() + right);
21         head = new ListNode(0);
22         ListNode* cur = head;
23         for (int i = 0; i < ans.size(); i++) {
24             cur->next = new ListNode(ans[i]);
25             cur = cur->next;
26         }
27         return head->next;
28     }
29 };
30
```

Testcase

Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

head =

[1,2,3,4,5]

left =

2

right =

4

Output

[1,4,3,2,5]

Expected

[1,4,3,2,5]

Code

C++

```
/**
 * 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:
    vector<int> reverseBetween(ListNode* head, int left, int right) {
        if (!head) return {};
        vector<int> ans;
        while (head) {
            ans.push_back(head->val);
            head = head->next;
        }
        reverse(ans.begin() + left - 1, ans.begin() + right);
        head = new ListNode(0);
        ListNode* cur = head;
        for (int i = 0; i < ans.size(); i++) {
            cur->next = new ListNode(ans[i]);
            cur = cur->next;
        }
        return head->next;
    }
};
```

View more

More challenges