

Leetcode-Rotate list

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
int GetLength(struct ListNode* head)
{
    if (head == NULL)
        return 0;

    return 1 + GetLength(head->next);
}

struct ListNode* rotateRight(struct ListNode* head, int k) {
    if (head == NULL || k == 0)
        return head;

    int length = GetLength(head);

    if (length == 1)
        return head;

    for(int i=0; i<k%length; i++)
    {
        struct ListNode *p=head;
        while(p->next->next!=NULL)
        {
            p=p->next;
        }

        struct ListNode *a=(struct ListNode *)malloc(sizeof(struct ListNode));
        a->val=p->next->val;
        a->next=head;
        head=a;
        p->next=NULL;
    }

    return head;
}
```

```
}
```

Accepted Runtime: 6 ms

- Case 1
- Case 2

Input

head =
[1,2,3,4,5]

k =
2

Output

[4,5,1,2,3]

Expected

[4,5,1,2,3]

Accepted Runtime: 6 ms

• Case 1

• Case 2

Input

head =
[0,1,2]

k =
4


Output

[2,0,1]

Expected

[2,0,1]

Accepted

 Rushila submitted at Feb 22, 2024 20:29

 Editorial

 Solution


 Runtime

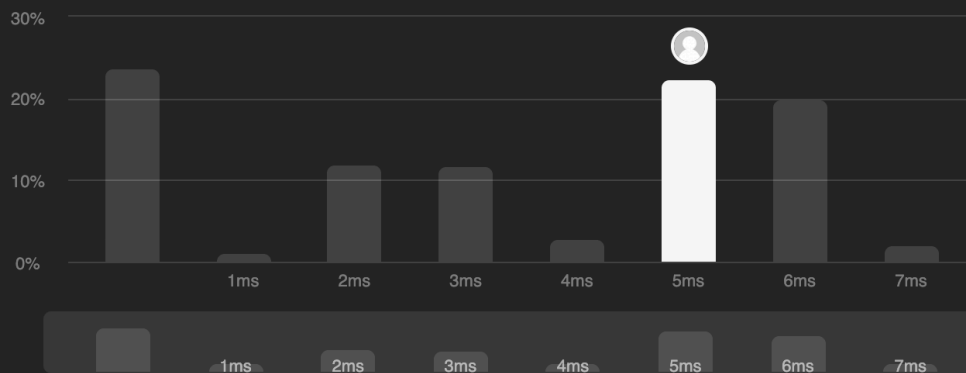
5 ms

Beats 49.52% of users with C

 Memory

6.24 MB

 Beats 50.60% of users with C



Code | C

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
int GetLength(struct ListNode* head)
```

 View more