# 题目

实现一种算法，找出单向链表中倒数第 k 个节点。返回该节点的值。

注意：本题相对原题稍作改动

**示例：**

输入： 1->2->3->4->5 和 k = 2

输出： 4

**说明：**

给定的 k 保证是有效的。

类似题目：剑指offer 22（返回值不同）

# 分析

## 方法一：双指针

**思路：**

**代码：**

/\*\*

\* Definition for singly-linked list.

\* struct ListNode {

\* int val;

\* ListNode \*next;

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

\* };

\*/

class Solution {

public:

int kthToLast(ListNode\* head, int k) {

ListNode \*former = head;

ListNode \*latter = head;

for(int i=0;i<k;i++)

{

former = former->next;

}

while(former)

{

former = former->next;

latter = latter->next;

}

return latter->val;

}

};

复杂度分析：

时间复杂度O(N) ：N为链表长度；总体看，former走了N步，latter走了(N-k)步。

空间复杂度O(1) ：双指针former , latter使用常数大小的额外空间。