

面试题22：链表中倒数第k个结点

题目描述

输入一个链表，输出该链表中倒数第k个结点。

解题思路

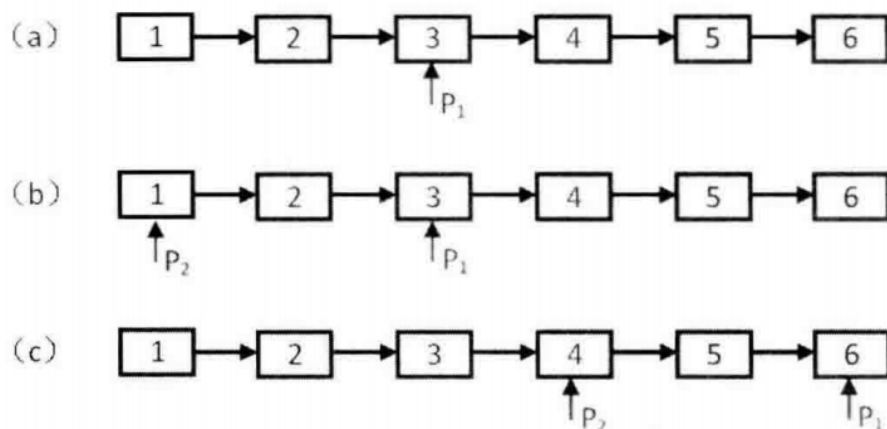


图 3.7 在有 6 个节点的链表上找倒数第 3 个节点的过程

注：(a) 第一个指针在链表上走两步。(b) 把第二个指针指向链表的头节点。(c) 两个指针一同沿着链表向前走。当第一个指针指向链表的尾节点时，第二个指针指向倒数第 3 个节点。

C++

```
1  /*
2  struct ListNode {
3      int val;
4      struct ListNode *next;
5      ListNode(int x) :
```

```

6         val(x), next(NULL) {
7     }
8 };*/
9 class Solution {
10 public:
11     ListNode* FindKthToTail(ListNode* pListHead,
12     unsigned int k) {
13         //为了实现只遍历链表一次就能找到倒数第k个节
14         点，我们可以定义两个指针
15         //①判断头指针为nullptr; ②k输入0，没有意义
16         if(pListHead == nullptr || k == 0)
17             return nullptr;
18         //定义两个指针
19         ListNode *pHead = pListHead;
20         ListNode *pTail = nullptr;
21         //第一个指针从链表的头指针开始遍历向前走k-1
22         步，第二个指针保持不动
23         for(unsigned int i = 0; i < k - 1; ++i){
24             //如果链表的节点数少于k
25             if(pHead -> next != nullptr)
26                 pHead = pHead -> next;
27             else{
28                 return nullptr;
29             }
30         }
31         //从第k步开始，第二个指针也开始从链表的头指
32         针开始遍历
33         pTail = pListHead;

```

```

30      //由于两个指针的距离保持在k-1，当第一个（走
    在前面的）指针到达链表的尾节点时，第二个（走在后面的）指针正好指向倒数第k个节点
31      while(pHead -> next != nullptr){
32          pHead = pHead -> next;
33          pTail = pTail -> next;
34      }
35      return pTail;
36  }
37  };

```

Java

```

1  /*
2  public class ListNode {
3      int val;
4      ListNode next = null;
5
6      ListNode(int val) {
7          this.val = val;
8      }
9  }*/
10 public class Solution {
11     public ListNode FindKthToTail(ListNode
head,int k) {
12         if(head==null){
13             return null;

```

```
14         }
15         int count=1;
16         ListNode old=head;
17         while(head.next!=null){
18             head=head.next;
19             count++;
20         }
21         if(k>count){
22             return null;
23         }
24         for(int i=0;i<count-k;i++){
25             old=old.next;
26         }
27         return old;
28     }
29 }
```