

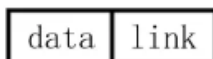
07、【2009】链表倒数第K个结点

24计算机考研成员一战成硕！



题目描述：

42. (15分) 已知一个带有表头结点的单链表，结点结构为



假设该链表只给出了头指针list。在不改变链表的前提下，请设计一个尽可能高效的算法，查找链表中倒数第k个位置上的结点（k为正整数）。若查找成功，算法输出该结点的data值，并返回1；否则，只返回0。要求：

- (1) 描述算法的基本设计思想
- (2) 描述算法的详细实现步骤
- (3) 根据设计思想和实现步骤，采用程序设计语言描述算法（使用C或C++或JAVA语言实现），关键之处请给出简要注释。

⌚ 倒计时

1、知识点及难度



题解人：酒客

难度：中等

知识点：利用双指针求解链表问题

2、算法题

思路 先让快指针走k个单位，然后我们再让快慢指针一起走，

是不是当快指针指向最后一个节点的时候慢指针就刚好指到倒数第k个节点了？

函数接收一个链表头节点和一个整数k作为参数。函数的返回值是删除节点后的链表头节点。

函数的实现逻辑如下：

1. 首先判断头节点是否为空，如果为空则直接返回头节点。
2. 创建两个指针fast和slow，都指向头节点。还创建一个指针preslow，初始值为空。
3. 让快指针fast先向后移动k个节点。
4. 接下来让快指针fast和慢指针slow一起向后移动，直到快指针fast指向最后一个节点。
5. 在移动的过程中，用preslow指针来记录慢指针slow的前一个节点，以便后面删除节点时使用。
6. 如果要删除的节点是头节点，即preslow为空，那么将头节点向后移动一个位置，并删除原来的头节点。
7. 如果要删除的节点不是头节点，那么只需要让preslow的下一个节点指向slow的下一个节点，然后删除slow节点。
8. 最后返回删除节点后的链表头节点。

整体思路是通过快慢指针的方式找到倒数第k个节点，并删除该节点

1.

基本实现-C++

```
1 ListNode* FindNode(ListNode* head,int k) {
2     if( !head )return head;
3     ListNode* fast = head;
4     ListNode* slow = head;
5     ListNode* preslow = nullptr;
6     //先让快指针走k个单位，然后我们再让快慢指针一起走，
7     //是不是当快指针指向最后一个节点的时候慢指针就刚好指到倒数第k个节点了？
8     while( k-- )fast = fast->next;
9     while( fast ) {
10         //preslow用来指向慢指针前一个元素方便删除，
11         //因为链表删除节点只能用前一个节点删除后面的哪个节点
12         preslow = slow;
13         slow = slow->next;
14         fast = fast->next;
15     }
16     //如果要删除的结点就是头节点，
17     //那此时preslow=null，只需让头节点向后移动，再删除原来的头节点
18     if(!preslow){
19         ListNode* pDel = head;
20         head = head->next;
```

```

21         delete pDel;
22     }
23     //如果要删除的不是头节点，那么只需要让被删除结点的前一个结点指向被删除结点的后一个结点
24     //即让preslow的下一个结点为slow的下一个结点，删除slow
25
26     else {
27         preslow->next = slow->next;
28         delete slow;
29     }
30     return head;
31 }ListNode* FindNode(ListNode* head,int k) {
32     if( !head )return head;
33     ListNode* fast = head;
34     ListNode* slow = head;
35     ListNode* preslow = nullptr;
36     //先让快指针走k个单位，然后我们再让快慢指针一起走，
37     //是不是当快指针指向最后一个节点的时候慢指针就刚好指到倒数第k个节点了？
38     while( k-- )fast = fast->next;
39     while( fast ) {
40         //preslow用来指向慢指针前一个元素方便删除，
41         //因为链表删除节点只能用前一个节点删除后面的哪个节点
42         preslow = slow;
43         slow = slow->next;
44         fast = fast->next;
45     }
46     //如果要删除的结点就是头节点，
47     //那此时preslow=null，只需让头节点向后移动，再删除原来的头节点
48     if(!preslow){
49         ListNode* pDel = head;
50         head = head->next;
51         delete pDel;
52     }
53     //如果要删除的不是头节点，那么只需要让被删除结点的前一个结点指向被删除结点的后一个结点
54     //即让preslow的下一个结点为slow的下一个结点，删除slow
55
56     else {
57         preslow->next = slow->next;
58         delete slow;
59     }
60     return head;
61 }

```

基本实现C

```

1 struct ListNode {

```

```

2     int val;
3     struct ListNode* next;
4 };
5
6 struct ListNode* FindNode(struct ListNode* head, int k)
7 {
8     if (!head)
9         return head;
10
11     struct ListNode* fast = head;
12     struct ListNode* slow = head;
13     struct ListNode* preslow = NULL;
14
15     // 让快指针先走k个单位
16     while (k--)
17         fast = fast->next;
18
19     // 快慢指针一起走
20     while (fast) {
21         preslow = slow;
22         slow = slow->next;
23         fast = fast->next;
24     }
25
26     // 如果要删除的结点是头结点
27     if (!preslow) {
28         struct ListNode* pDel = head;
29         head = head->next;
30         free(pDel);
31     }
32     // 如果要删除的不是头结点
33     else {
34         preslow->next = slow->next;
35         free(slow);
36     }
37
38     return head;
39 }

```

3、总结



总结栏

蓝蓝B站首页：[蓝蓝希望你上岸呀B站首页](#)

蓝蓝公众号：[算法训练营9分计划](#)

蓝蓝知识星球介绍：[👁 关于知识星球的权益](#)

如何在星球打卡记录：

- 计算机考研数据结构算法专项day[1/60]：
- 学习内容：最好能发出自己写的图片
- 遇到的问题：如果无就不用写了
- 小结：这部分一周写一次即可。