

[LeetCode] Remove Duplicates from Sorted List - 链表问题

原创 Eastmount 2015-09-10 03:33:30 2140 收藏

版权

分类专栏: LeetCode 文章标签: leetcode 链表



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法、神...



Eastmount

¥9.90

订阅博主

题目概述:

Given a sorted linked list, delete all duplicates such that each element appear only once.

For example,

Given 1->1->2, return 1->2.

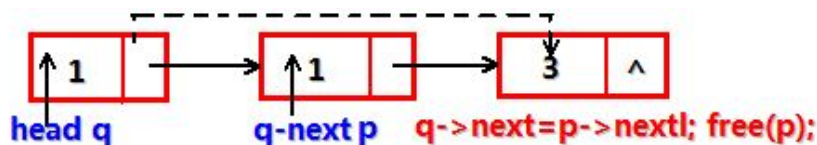
Given 1->1->2->3->3, return 1->2->3.

题目解析:

这是一道非常简单的链表题目，题意是删除单链表(已排序)中的重复数字，只需一次判断前后两个结点数字是否相等即可。需要注意几点：

- 1.该链表中头结点就开始存储数字head->val存在；
- 2.初始判断if(head==NULL || head->next==NULL)，防止出现'[]'或'[1]';
- 3.判断过程中使用q->next->val==q->val和释放free临时结点p；
- 4.若使用中间变量number=q->val判断时，当else中q指向下一个结点为空时，该句不存在number=q->val会报错RE，如'[1,1]'。故不建议使用临时变量。

总之，一道非常基础的链表题目，不需要过于复杂化代码。



我的代码:

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
```

```

*      int val;
*      struct ListNode *next;
* };
*/
struct ListNode* deleteDuplicates(struct ListNode* head) {
    struct ListNode *p,*q;
    q=head;
    if(head==NULL || head->next==NULL)        //防止[]和[1]
        return head;
    while(q) {
        if( q->next!=NULL && q->next->val==q->val ) {
            //删除操作 最后free
            p=q->next;
            q->next=p->next;
            free(p);
        }
        else {
            q=q->next;
        }
    }
    return head;
}

```

其他类型链表题目：

(By:Eastmount 2015-9-10 凌晨3点半 <http://blog.csdn.net/eastmount/>)