

# [LeetCode] Reverse Linked List I II - 链表翻转问题

原创

Eastmount

2015-09-14 19:09:56

2560

☆ 收藏

版权

分类专栏:

LeetCode

文章标签:

leetcode

链表

翻转



## 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:

Reverse a singly linked list.

翻转一个单链表，如：1->2 输出 2->1；1->2->3 输出3->2->1。

### 题目解析:

本人真的比较笨啊！首先想到的方法就是通过判断链尾是否存在，再新建一个链表，每次移动head的链尾元素，并删除head链表中的元素，一个字“蠢”，但好歹AC且巩固了链表基础知识。你可能遇见的错误包括：

1. 'ListNode' undeclared (first use in this function)

```
nhead=(istNode*)malloc(sizeof(ListNode));
```

=>

```
nhead=(struct ListNode*)malloc(sizeof(struct ListNode));
```

2. Time Limit Exceeded

在链表遍历寻找最后一个结点并插入新链表尾部中需要注意，建议的方法：

```
q=head; while(q) {q=q->next;}
```

```
p=(struct ListNode*)malloc(sizeof(struct ListNode));
```

```
p->val=head->val; p->next=NULL; q=p;
```

=>

```
q=head; while(q) {last=q; q=q->next;}
```

```
p=(struct ListNode*)malloc(sizeof(struct ListNode));
```

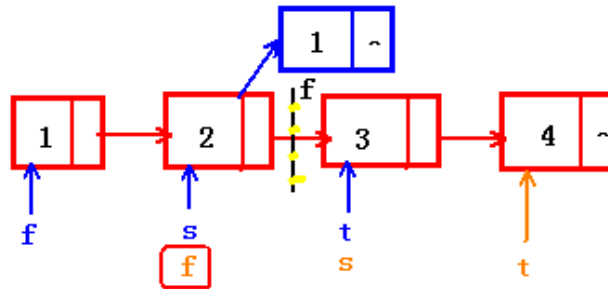
```
p->val=head->val; p->next=NULL; last->next=p;
```

通过借助last变量更直观，否则结果总是错误。而且此时q为next指向NULL，如果用到q->next=p就会出现RE错误，因为q都为NULL，哪来的q->next。第二个错误也可能是我个人的编程习惯吧！

第二种方法更为推荐——直接翻转，还有一种递归方法自行提高。

如下图所示，红色表示初始链表存在4个值[1, 2, 3, 4]，蓝色表示初始指针first指向

第一个元素、second指向第二个元素(head->next), third指向第三个元素; 首先 s->next=f断开链表并翻转指向第一个元素, 同时f=s最后返回first。如果只有两个元素 [1, 2]则执行"s->next=f; f=s;"后s=t=NULL返回f即可输出[2, 1]。



## 我的代码:

### 直接翻转方法

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
struct ListNode* reverseList(struct ListNode* head) {
    struct ListNode *first,*second,*third;
    if(head==NULL||head->next==NULL)
        return head;
    first = head;
    second = head->next;
    first->next = NULL;
    while(second!=NULL) { //注意while(second) 不能执行
        third = second->next;
        second->next = first;
        first = second;
        second = third;
    }
    return first;
}
```

### "蠢"方法

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