

# 字节跳动的面试算法题，实在太变态了

开发者技术前线 2019-11-19

以下文章来源于苦逼的码农，作者帅地



苦逼的码农

本号专注于讲解数据结构与算法、计算机基础(如计算机网络+操作系统+数据库+Linux)...



点击“[开发者技术前线](#)”，选择“星标”

13: 21 在看|星标|留言，真爱



来源公众号：苦逼的码农 编辑：可可

前几天我去面试字节跳动，面试官问了一道链表相关的算法题，不过我第一时之间没做出来，就回来看了一下，感觉这道题还不错，拿来讲一讲。

## 题目

这其实是一道变形的链表反转题，大致描述如下

给定一个单链表的头节点 head,实现一个调整单链表的函数，使得每K个节点之间为一组进行逆序，并且从链表的尾部开始组起，头部剩余节点数量不够一组的不需要逆序。  
(不能使用队列或者栈作为辅助)

例如：

链表:1->2->3->4->5->6->7->8->null, K = 3。那么 6->7->8, 3->4->5, 1->2各位

一组。调整后：1->2->5->4->3->8->7->6->null。其中 1，2 不调整，因为不够一组。

## 解答

这道题的难点在于，是从链表的尾部开始组起的，而不是从链表的头部，如果是头部的话，那我们还是比较容易做的，因为你可以遍历链表，每遍历 k 个就拆分为一组来逆序。但是从尾部的话就不一样了，因为是单链表，不能往后遍历组起。不过这道题肯定是用递归比较好做，对递归不大懂的建议看我之前写的一篇文章[身为技术专家的我，面试居然还要靠刷题?](#)这篇文章写了关于递归的一些套路。

## 先做一道类似的反转题

在做这道题之前，我们不仿先来看看[如果从头部开始组起的话](#)，应该怎么做呢？例如：链表:1->2->3->4->5->6->7->8->null, K = 3。调整后：3->2->1->6->5->4->7->8->null。其中 7，8 不调整，因为不够一组。

这道题我们可以用递归来实现，假设方法reverseKNode()的功能是将单链表的每K个节点之间逆序(从头部开始组起的哦);reverse()方法的功能是将一个单链表逆序。

那么对于下面的这个单链表，其中 K = 3。



我们把前K个节点与后面的节点分割出来：



temp指向的剩余的链表，可以说是原问题的一个子问题。  
我们可以调用reverseKNode()方法将temp指向的链表每K个节点之间进行逆序。  
再调用reverse()方法把head指向的那3个节点进行逆序，结果如下：



再次声明，如果对这个递归看不大懂的，建议看下我那篇递归的文章

接着，我们只需要把这两部分给连接起来就可以了。最后的结果如下：



代码如下：

```
//k个为一组逆序
public ListNode reverseKGroup(ListNode head, int k) {
    ListNode temp = head;
```

```

        for (int i = 1; i < k && temp != null; i++) {
            temp = temp.next;
        }
        //判断节点的数量是否能够凑成一组
        if(temp == null)
            return head;

        ListNode t2 = temp.next;
        temp.next = null;
        //把当前的组进行逆序
        ListNode newHead = reverseList(head);
        //把之后的节点进行分组逆序
        ListNode newTemp = reverseKGroup(t2, k);
        // 把两部分连接起来
        head.next = newTemp;

        return newHead;
    }

    //逆序单链表
    private static ListNode reverseList(ListNode head) {
        if(head == null || head.next == null)
            return head;
        ListNode result = reverseList(head.next);
        head.next.next = head;
        head.next = null;
        return result;
    }
}

```

## 回到本题

这两道题可以说是及其相似的了，只是一道从头部开始组起，这道从头部开始组起的，也是 leetcode 的第 25 题。而面试的时候，经常会进行变形，例如这道字节跳动的题，它变成从**尾部**开始组起，可能你一时之间就不知道该怎么弄了。当然，可能有人一下子就反应出来，把他秒杀了。

其实这道题很好做滴，你只需要先把单链表进行一次**逆序**，逆序之后就能转化为**从头部开始组起**了，然后按照我上面的解法，处理完之后，把结果**再次逆序**即搞定。两次逆序相当于没逆序。

例如对于链表（其中  $K = 3$ ）



我们把它从尾部开始组起，每  $K$  个节点为一组进行逆序。

步骤如下：

### 1、先进行逆序



逆序之后就可以把问题转化为从头部开始组起，每  $K$  个节点为一组进行逆序。

### 2、处理后的结果如下



### 3、接着在把结果逆序一次，结果如下



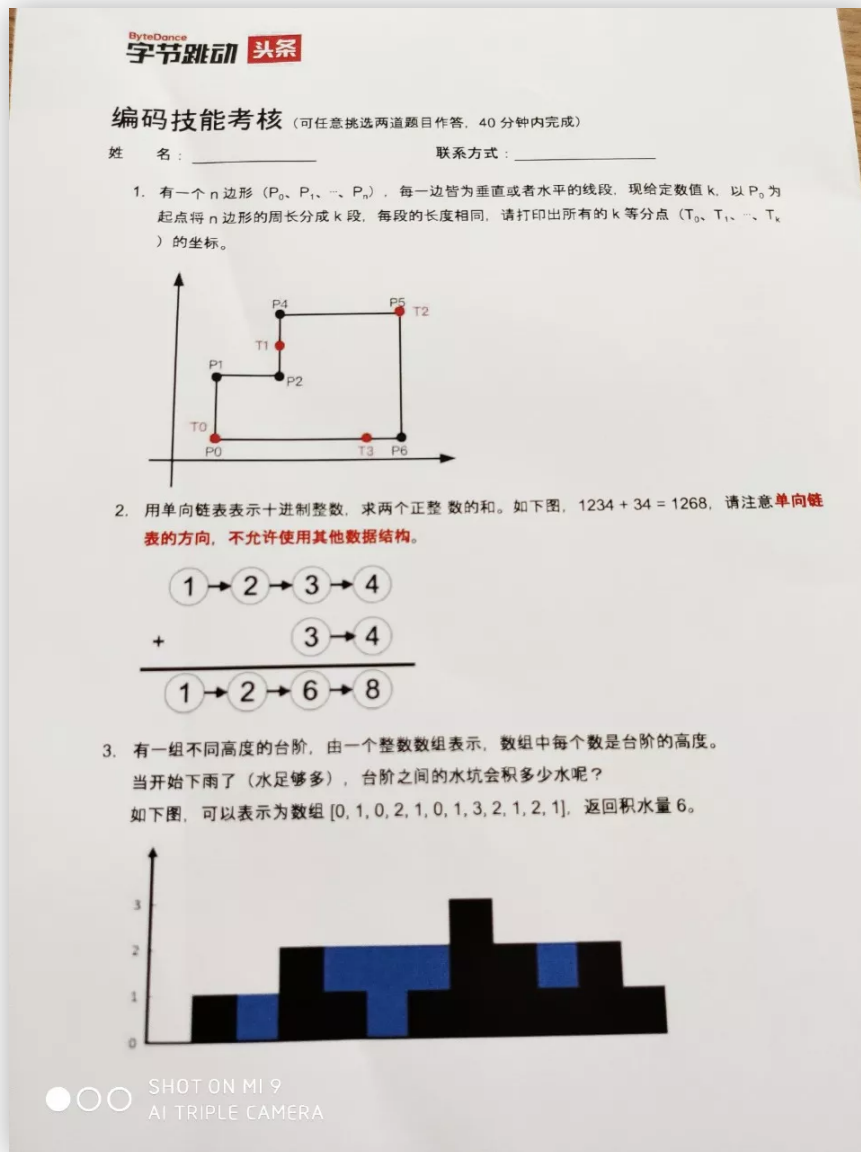
代码如下：

```
public ListNode solve(ListNode head, int k) {  
    // 调用逆序函数  
    head = reverse(head);  
    // 调用每 k 个为一组的逆序函数（从头部开始组起）  
    head = reverseKGroup(head, k);  
    // 在逆序一次  
    head = reverse(head);  
}
```

```
return head;
```

```
}
```

类似于这种需要先进行逆序的还要两个链表相加，这道题字节跳动的笔试题也有出过，如下图的第二题：



这道题就需要先把两个链表逆序，再节点间相加，最后在合并了。

## 总结

关于链表的算法题，在面试的时候听说是挺常考的，大家可以多注意注意，遇到不错的链表算法题，可以看身为技术专家的我，面试居然还要靠刷题？

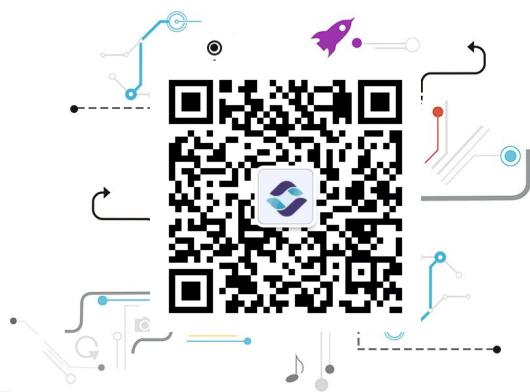
开发者技术前线，汇集技术前线快讯和关注行业趋势，大厂干货，是开发者经历和成长的优秀指南。

## 历史推荐

苹果正在试图杀死 Web 技术

使用 Flutter 之后，CPU 占用率降了 50%

不要自称程序员



好文点个在看吧！