**难度: 简单**

实现一种算法，删除单向链表中间的某个节点（即不是第一个或最后一个节点），假定你只能访问该节点。

示例：

输入：单向链表a->b->c->d->e->f中的节点c

结果：不返回任何数据，但该链表变为a->b->d->e->f

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/delete-middle-node-lcci

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路:**

1). 首先它**这个结点就是要删除的结点(node)**

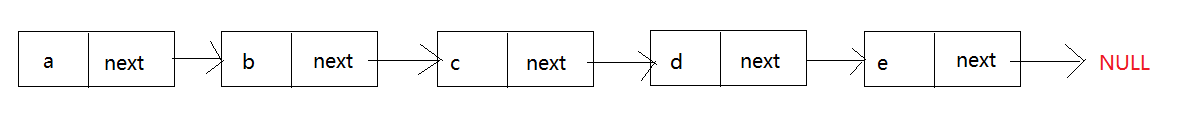
(刚开始还没注意这点 ,我以为这个(node)是第一个结点,然后在总的结点中,删除中间的结点)

|  |
| --- |
| public void deleteNode(ListNode node) {    } |

2). 那我们先让这个node的值,换成它的下一个值

即:

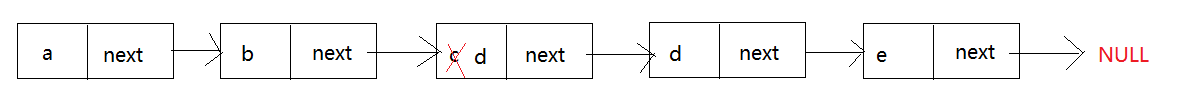
1. 一开始



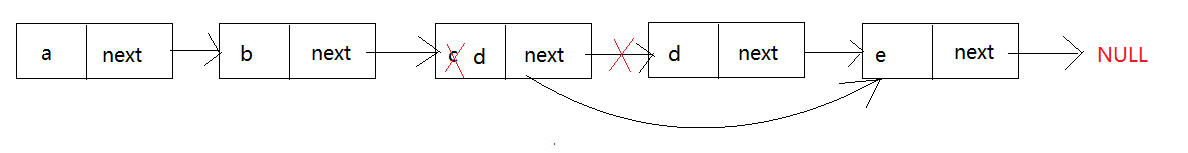
2. 现在我们替换这个需要删除c

(此时node就是c点)

即把c点换成d



3). 然后我们只需要让替换成d的这个点**指向它下个结点**的**下一个结点**



1. 此时**真正的d点**由于没有结点指向它,过一段时间会被垃圾回收机制自动回收

代码:下面

|  |
| --- |
| public void deleteNode(ListNode node) {  node.val = node.next.val;  node.next = node.next.next;  } |