赋值法

复杂度

时间 O(1) 空间 O(1)

思路

正常情况下链表中节点的删除是需要知道被删除节点的前一个节点的，将它前一个节点的next指针指向它的下一个节点，这个节点就从链表中删除了。但是这里没有提供前一个节点，而是提供了当前节点。

乍一看没法获取上一个链表节点，似乎无法将当前结点去除。实际上只要将下一个节点的值覆盖当前节点，然后删除下一个节点就好了。但是需要注意这种方法不能删除尾节点（题目中也给出了这个条件）

代码

1. **void** deleteNode(ListNode\* node) {
2. **if** (!node || !node->next)
3. **return**;
4. node->val = node->next->val;
5. node->next = node->next->next;
7. }

甚至可以直接

void deleteNode(ListNode\* node) {

\*node = \*(node->next);

}

用要删除的结点的下一个节点把要删除的结点覆盖掉。

if the node should be freed, we need 3 lines:

**class** **Solution** {

**public**:

**void** deleteNode(ListNode\* node) {

ListNode\* temp = node->next;

\*node = \*temp;

delete temp;

}

};