

```

1  /*
2  请编写一个函数，使其可以删除某个链表中给定的（非末尾）节点，你将只被给定要求被删除的节点。
3
4  现有一个链表 -- head = [4,5,1,9]，它可以表示为：
5  4-->5-->1-->9
6  示例 1：
7  输入：head = [4,5,1,9]，node = 5
8  输出：[4,1,9]
9  解释：给定你链表中值为 5 的第二个节点，那么在调用了你的函数之后，该链表应变为 4 -> 1 -> 9.
10 示例 2：
11
12 输入：head = [4,5,1,9]，node = 1
13 输出：[4,5,9]
14 解释：给定你链表中值为 1 的第三个节点，那么在调用了你的函数之后，该链表应变为 4 -> 5 -> 9.
15
16 说明：
17
18     链表至少包含两个节点。
19     链表中所有节点的值都是唯一的。
20     给定的节点为非末尾节点并且一定是链表中的一个有效节点。
21     不要从你的函数中返回任何结果。
22
23 来源：力扣（LeetCode）
24 链接：https://leetcode-cn.com/problems/delete-node-in-a-linked-list
25 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。
26 */

```

分析:

- 从头开始遍历链表(需要两个指针,一个负责遍历,一个指向当前节点的前驱节点),当遍历到符合条件的节点时,先保存当前节点的指针到`temp`,然后让当前节点的前驱的`next`指向当前节点的`next`,再释放`temp`.(看错题目了)
- 阅读理解:输入节点其实是要删除的节点,由于没有传入前驱节点,那就只能从当前节点开始从后面往前面进行`val`值的覆盖,然后释放最后一个节点,并让倒数第二个节点的`next`指向NULL了.

方法一: C\_滑动窗遍历

```

1  /**
2   * Definition for singly-linked list.
3   * struct ListNode {
4   *     int val;
5   *     struct ListNode *next;
6   * };
7   */
8  void deleteNode(struct ListNode* node)
9  {
10     struct ListNode* front = node ;
11     struct ListNode* cur = node->next ;
12
13     while(cur->next)

```

```
14     {
15         front->val = cur->val ;
16         front     = cur      ;
17         cur       = cur->next ;
18     }
19
20     front->val = cur->val ;
21     free(front->next)    ;
22     front->next = NULL   ;
23
24     return ;
25 }
26
27 /*
28 执行结果:
29 通过
30 显示详情
31 执行用时 :8 ms, 在所有 C 提交中击败了87.65% 的用户
32 内存消耗 :7.7 MB, 在所有 C 提交中击败了5.09%的用户
33 */
```

---

AlimyBreak  
2019.08.22