# 18.1 在 O(1) 时间内删除链表节点

## 解题思路

① 如果该节点不是尾节点，那么可以直接将下一个节点的值赋给该节点，然后令该节点指向下下个节点，再删除下一个节点，时间复杂度为 O(1)。

② 否则，就需要先遍历链表，找到节点的前一个节点，然后让前一个节点指向 null，时间复杂度为 O(N)。

综上，如果进行 N 次操作，那么大约需要操作节点的次数为 N-1+N=2N-1，其中 N-1 表示 N-1 个不是尾节点的每个节点以 O(1) 的时间复杂度操作节点的总次数，N 表示 1 个尾节点以 O(N) 的时间复杂度操作节点的总次数。(2N-1)/N ~ 2，因此该算法的平均时间复杂度为 O(1)。

public ListNode deleteNode(ListNode head, ListNode tobeDelete) {
 if (head == null || tobeDelete == null)
 return null;
 if (tobeDelete.next != null) {
 // 要删除的节点不是尾节点
 ListNode next = tobeDelete.next;
 tobeDelete.val = next.val;
 tobeDelete.next = next.next;
 } else {
 if (head == tobeDelete)
 // 只有一个节点
 head = null;
 else {
 ListNode cur = head;
 while (cur.next != tobeDelete)
 cur = cur.next;
 cur.next = null;
 }
 }
 return head;
}