Problem 237: Delete Node in a Linked List

Problem Information

Difficulty: Medium
Acceptance Rate: 83.21%
Paid Only: No
Tags: Linked List

Problem Description

There is a singly-linked list `head` and we want to delete a node `node` in it.

You are given the node to be deleted `node`. You will **not be given access** to the first node of `head`.

All the values of the linked list are **unique** , and it is guaranteed that the given node `node` is not the last node in the linked list.

Delete the given node. Note that by deleting the node, we do not mean removing it from memory. We mean:

* The value of the given node should not exist in the linked list. * The number of nodes in the linked list should decrease by one. * All the values before `node` should be in the same order. * All the values after `node` should be in the same order.

Custom testing:

* For the input, you should provide the entire linked list `head` and the node to be given `node`. `node` should not be the last node of the list and should be an actual node in the list. * We will build the linked list and pass the node to your function. * The output will be the entire list after calling your function.

Example 1:

![](https://assets.leetcode.com/uploads/2020/09/01/node1.jpg)

Input: head = [4,5,1,9], node = 5 **Output:** [4,1,9] **Explanation:** You are given the second node with value 5, the linked list should become 4 -> 1 -> 9 after calling your function.

Example 2:

![](https://assets.leetcode.com/uploads/2020/09/01/node2.jpg)

Input: head = [4,5,1,9], node = 1 **Output:** [4,5,9] **Explanation:** You are given the third node with value 1, the linked list should become 4 -> 5 -> 9 after calling your function.

Constraints:

* The number of the nodes in the given list is in the range `[2, 1000]`. * `-1000 <= Node.val <= 1000` * The value of each node in the list is **unique**. * The `node` to be deleted is **in the list** and is **not a tail** node.

Code Snippets

C++:

```
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode(int x) : val(x), next(NULL) {}
* };
*/
class Solution {
public:
void deleteNode(ListNode* node) {

}
};
```

Java:

```
/**
* Definition for singly-linked list.
* public class ListNode {
```

```
* int val;
* ListNode next;
* ListNode(int x) { val = x; }
* }
*/
class Solution {
public void deleteNode(ListNode node) {

}
}
```

Python3:

```
# Definition for singly-linked list.
# class ListNode:
# def __init__(self, x):
# self.val = x
# self.next = None

class Solution:
def deleteNode(self, node):
"""
:type node: ListNode
:rtype: void Do not return anything, modify node in-place instead.
"""
```