Problem 2487: Remove Nodes From Linked List

Problem Information

Difficulty: Medium
Acceptance Rate: 74.66%
Paid Only: No
Tags: Linked List, Stack, Recursion, Monotonic Stack

Problem Description

You are given the `head` of a linked list.

Remove every node which has a node with a greater value anywhere to the right side of it.

Return _the_`head` _of the modified linked list._

Example 1:

Input: head = [5,2,13,3,8] **Output:** [13,8] **Explanation:** The nodes that should be removed are 5, 2 and 3. - Node 13 is to the right of node 5. - Node 13 is to the right of node 2. - Node 8 is to the right of node 3.

Example 2:

Input: head = [1,1,1,1] **Output:** [1,1,1,1] **Explanation:** Every node has value 1, so no nodes are removed.

Constraints:

* The number of the nodes in the given list is in the range `[1, 105]`. * `1 <= Node.val <= 105`

Code Snippets

C++:

```
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode() : val(0), next(nullptr) {}
* ListNode(int x) : val(x), next(nullptr) {}
* ListNode(int x, ListNode *next) : val(x), next(next) {}
* };
*/
class Solution {
public:
ListNode* removeNodes(ListNode* head) {

}
};
```

Java:

```
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
* ListNode() {}
* ListNode(int val) { this.val = val; }
* ListNode(int val, ListNode next) { this.val = val; this.next = next; }
* }
*/
class Solution {
public ListNode removeNodes(ListNode head) {

}
}
```

Python3:

```
# Definition for singly-linked list.
# class ListNode:
# def __init__(self, val=0, next=None):
# self.val = val
```

```
# self.next = next
class Solution:
def removeNodes(self, head: Optional[ListNode]) -> Optional[ListNode]:
```