Problem 3217: Delete Nodes From Linked List Present in Array

Problem Information

Difficulty: Medium
Acceptance Rate: 69.69%
Paid Only: No
Tags: Array, Hash Table, Linked List

Problem Description

You are given an array of integers `nums` and the `head` of a linked list. Return the `head` of the modified linked list after **removing** all nodes from the linked list that have a value that exists in `nums`.

Example 1:

Input: nums = [1,2,3], head = [1,2,3,4,5]

Output: [4,5]

Explanation:

Remove the nodes with values 1, 2, and 3.

Example 2:

Input: nums = [1], head = [1,2,1,2,1,2]

Output: [2,2,2]

Explanation:

Remove the nodes with value 1.

Example 3:

Input: nums = [5], head = [1,2,3,4]

Output: [1,2,3,4]

Explanation:

No node has value 5.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 105` * All elements in `nums` are unique. * The number of nodes in the given list is in the range `[1, 105]`. * `1 <= Node.val <= 105` * The input is generated such that there is at least one node in the linked list that has a value not present in `nums`.

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* modifiedList(vector<int>& nums, 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 modifiedList(int[] nums, 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 modifiedList(self, nums: List[int], head: Optional[ListNode]) ->
Optional[ListNode]:
```