3217. Delete Nodes From Linked List Present in Array

Solved

Medium | Topics | Companies | Hint

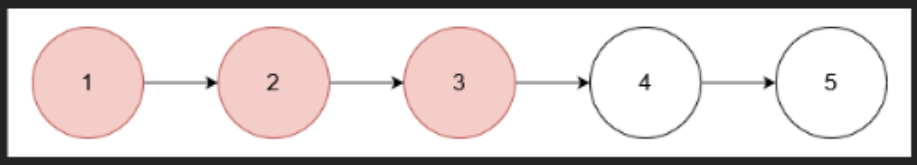
You are given an array of integers `nums` and the `head` of a linked list. Return the `head` of the modified linked list after **removing** all nodes from the linked list that have a value that exists in `nums`.

Example 1:

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

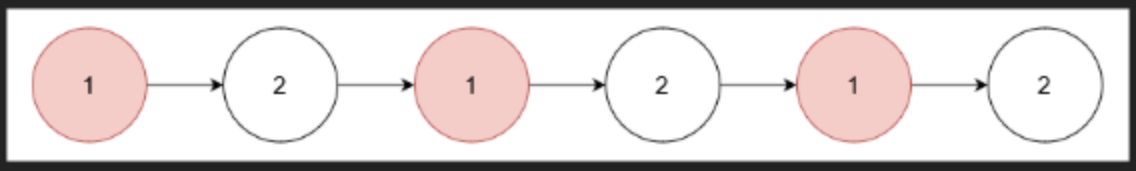
Output: `[4,5]`

Explanation:

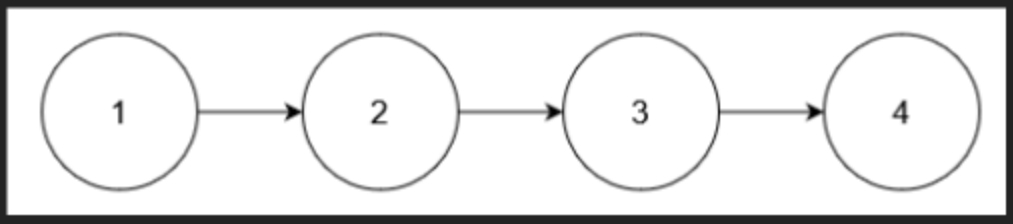


Remove the nodes with values 1, 2, and 3.

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:

- $1 \le nums.length \le 10^5$
- $1 \le nums[i] \le 10^5$
- All elements in `nums` are unique.
- The number of nodes in the given list is in the range $[1, 10^5]$.
- $1 \le Node.val \le 10^5$
- The input is generated such that there is at least one node in the linked list that has a value not present in `nums`.

Python:

```
# 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]:
        max_val = -1
        for num in nums:
            max_val = max(num, max_val)

        freq = [False] * (max_val + 1)

        for num in nums:
            freq[num] = True

        temp = ListNode()
        current = temp

        while head:
            if head.val >= len(freq) or not freq[head.val]:
                current.next = head
                current = current.next
            head = head.next

        current.next = None
```

```
        return temp.next
```

JavaScript:

```
/**
 * Definition for singly-linked list.
 * function ListNode(val, next) {
 *     this.val = (val===undefined ? 0 : val)
 *     this.next = (next===undefined ? null : next)
 * }
 */
/**
 * @param {number[]} nums
 * @param {ListNode} head
 * @return {ListNode}
 */
var modifiedList = function(nums, head) {
    let max = -1;
    for( let num of nums ){
        max = num > max ? num : max;
    }

    let freq = new Array(max+1).fill(false);

    for(let num of nums)freq[num] = true;

    let temp = new ListNode();
    let current = temp;

    while(head != null ){
        if( head.val >= freq.length || freq[head.val] == false){
            current.next = head;
            current = current.next;
        }
        head = head.next;
    }

    current.next = null;
    return temp.next;
};
```

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) {
        int max = -1;
        for(int num : nums ){
            max = num > max ? num : max;
        }
        boolean[] freq = new boolean[max+1];

        for(int num : nums) freq[num] = true;

        ListNode temp = new ListNode();
        ListNode current = temp;

        while(head != null){
            if( head.val >= freq.length || freq[head.val] == false){
                current.next = head;
                current = current.next;
            }
            head = head.next;
        }

        current.next = null;
        return temp.next;
    }
}
```