Problem 3263: Convert Doubly Linked List to Array I

Problem Information

Difficulty: Easy
Acceptance Rate: 94.81%
Paid Only: Yes
Tags: Array, Linked List, Doubly-Linked List

Problem Description

You are given the `head` of a **doubly linked list** , which contains nodes that have a next pointer and a previous pointer.

Return an integer array which contains the elements of the linked list **in order**.

Example 1:

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

Output: [1,2,3,4,3,2,1]

Example 2:

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

Output: [2,2,2,2,2]

Example 3:

Input: head = [3,2,3,2,3,2]

Output: [3,2,3,2,3,2]

Constraints:

* The number of nodes in the given list is in the range `[1, 50]`. * `1 <= Node.val <= 50`

Code Snippets

C++:

```
/**
* Definition for doubly-linked list.
* class Node {
* int val;
* Node* prev;
* Node* next;
* Node() : val(0), next(nullptr), prev(nullptr) {}
* Node(int x) : val(x), next(nullptr), prev(nullptr) {}
* Node(int x, Node *prev, Node *next) : val(x), next(next), prev(prev) {}
* };
*/
class Solution {
public:
vector<int> toArray(Node *head){

}
};
```

Java:

```
/*
// Definition for a Node.
class Node {
public int val;
public Node prev;
public Node next;
};
*/

class Solution {
public int[] toArray(Node head) {

}
}
```

Python3:

```
"""
# Definition for a Node.
class Node:
def __init__(self, val, prev=None, next=None):
self.val = val
self.prev = prev
self.next = next
"""
class Solution:
def toArray(self, root: 'Optional[Node]') -> List[int]:
```