

Problem 3294: Convert Doubly Linked List to Array II

Problem Information

Difficulty: Medium

Acceptance Rate: 81.59%

Paid Only: Yes

Tags: Array, Linked List, Doubly-Linked List

Problem Description

You are given an **arbitrary** `node` from 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 = [1,2,3,4,5]`, `node = 5`

Output: `[1,2,3,4,5]`

Example 2:

Input: `head = [4,5,6,7,8]`, `node = 8`

Output: `[4,5,6,7,8]`

Constraints:

* The number of nodes in the given list is in the range `[1, 500]`. * `1 <= Node.val <= 1000` *
All nodes have unique `Node.val`.

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 *node){

    }

};
```

Java:

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

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

}

}
```

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, node: 'Optional[Node]') -> List[int]:
```