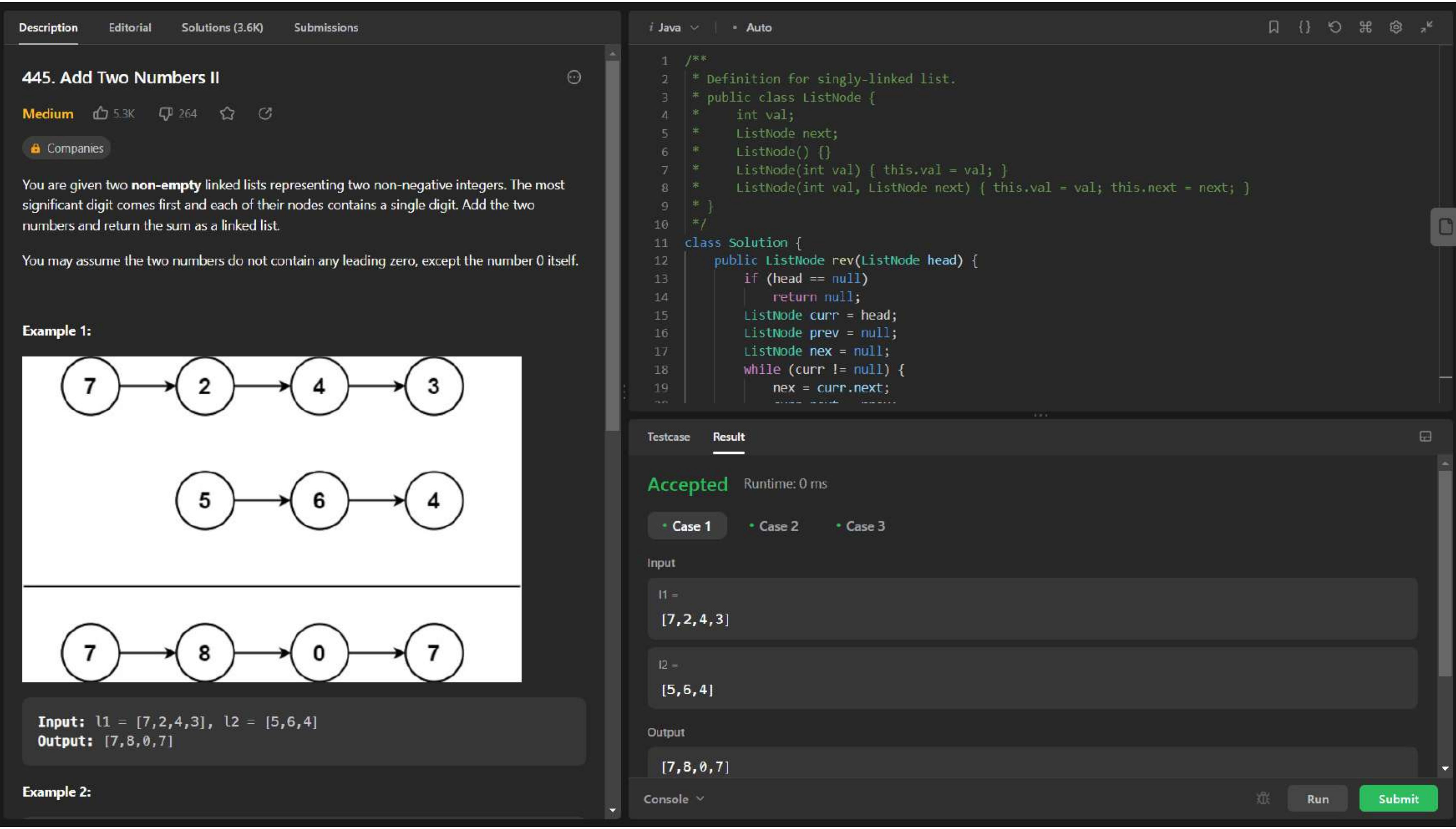


Description Editorial Solutions (3.6K) Submissions

445. Add Two Numbers II

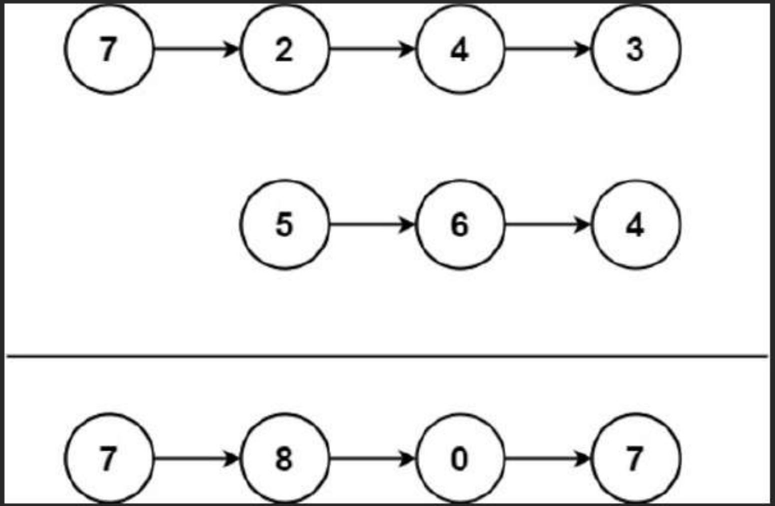
Medium 5.3K 264

Companies

You are given two **non-empty** linked lists representing two non-negative integers. The most significant digit comes first and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

Example 1:



```
Input: l1 = [7,2,4,3], l2 = [5,6,4]
Output: [7,8,0,7]
```

Example 2:

Java Auto

```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 rev(ListNode head) {
        if (head == null)
            return null;
        ListNode curr = head;
        ListNode prev = null;
        ListNode nex = null;
        while (curr != null) {
            nex = curr.next;
```

Testcase Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

l1 =

```
[7,2,4,3]
```

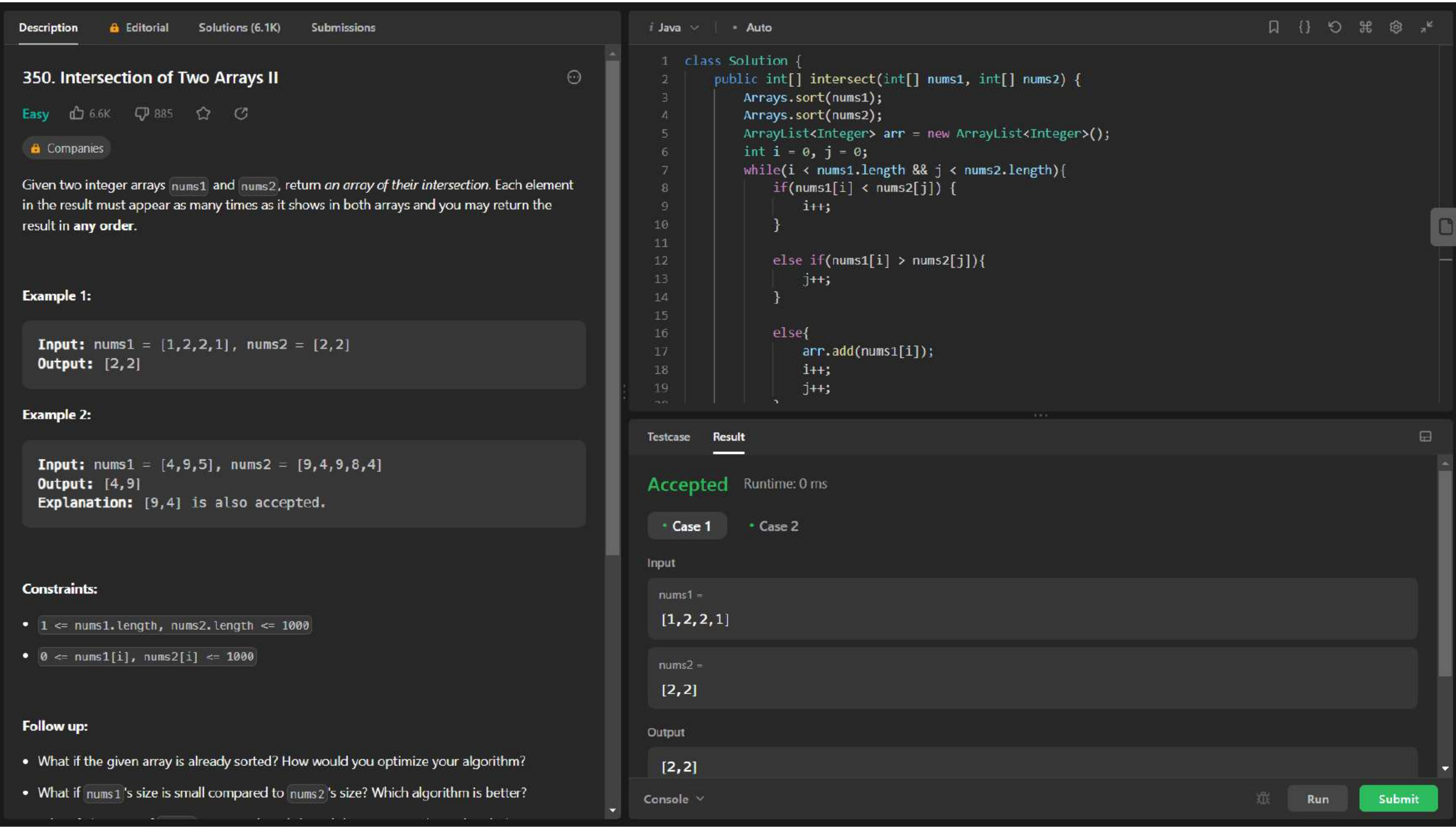
l2 =

```
[5,6,4]
```

Output

```
[7,8,0,7]
```

Console Run Submit



350. Intersection of Two Arrays II

Easy 6.6K 885

Companies

Given two integer arrays `nums1` and `nums2`, return *an array of their intersection*. Each element in the result must appear as many times as it shows in both arrays and you may return the result in **any order**.

Example 1:

```
Input: nums1 = [1,2,2,1], nums2 = [2,2]
Output: [2,2]
```

Example 2:

```
Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]
Output: [4,9]
Explanation: [9,4] is also accepted.
```

Constraints:

- `1 <= nums1.length, nums2.length <= 1000`
- `0 <= nums1[i], nums2[i] <= 1000`

Follow up:

- What if the given array is already sorted? How would you optimize your algorithm?
- What if `nums1`'s size is small compared to `nums2`'s size? Which algorithm is better?

```java
class Solution {
    public int[] intersect(int[] nums1, int[] nums2) {
        Arrays.sort(nums1);
        Arrays.sort(nums2);
        ArrayList<Integer> arr = new ArrayList<Integer>();
        int i = 0, j = 0;
        while(i < nums1.length && j < nums2.length){
            if(nums1[i] < nums2[j]) {
                i++;
            }

            else if(nums1[i] > nums2[j]){
                j++;
            }

            else{
                arr.add(nums1[i]);
                i++;
                j++;
```

Accepted Runtime: 0 ms

Case 1 Case 2

Input

```
nums1 =
[1,2,2,1]
nums2 =
[2,2]
```

Output

```
[2,2]
```