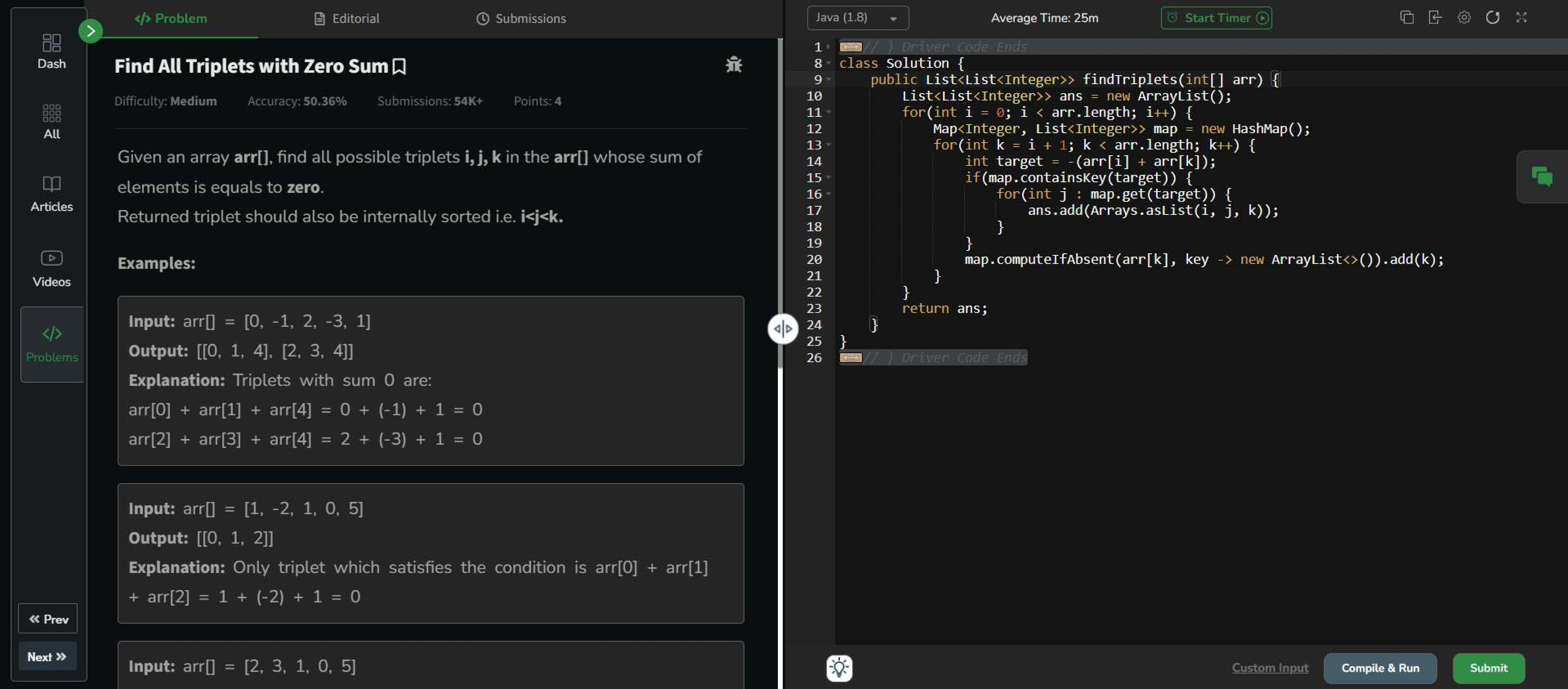

Find All Triplets with Zero Sum

Difficulty: **Medium** Accuracy: **50.36%** Submissions: **54K+** Points: **4**

Given an array **arr[]**, find all possible triplets **i, j, k** in the **arr[]** whose sum of elements is equals to **zero**.
Returned triplet should also be internally sorted i.e. **i<j<k.**

Examples:

Input: arr[] = [0, -1, 2, -3, 1]
Output: [[0, 1, 4], [2, 3, 4]]
Explanation: Triplets with sum 0 are:
arr[0] + arr[1] + arr[4] = 0 + (-1) + 1 = 0
arr[2] + arr[3] + arr[4] = 2 + (-3) + 1 = 0

Input: arr[] = [1, -2, 1, 0, 5]
Output: [[0, 1, 2]]
Explanation: Only triplet which satisfies the condition is arr[0] + arr[1] + arr[2] = 1 + (-2) + 1 = 0

Input: arr[] = [2, 3, 1, 0, 5]

```java
// } Driver Code Ends
class Solution {
    public List<List<Integer>> findTriplets(int[] arr) {
        List<List<Integer>> ans = new ArrayList();
        for(int i = 0; i < arr.length; i++) {
            Map<Integer, List<Integer>> map = new HashMap();
            for(int k = i + 1; k < arr.length; k++) {
                int target = -(arr[i] + arr[k]);
                if(map.containsKey(target)) {
                    for(int j : map.get(target)) {
                        ans.add(Arrays.asList(i, j, k));
                    }
                }
                map.computeIfAbsent(arr[k], key -> new ArrayList<>()).add(k);
            }
        }
        return ans;
    }
}
// } Driver Code Ends
```