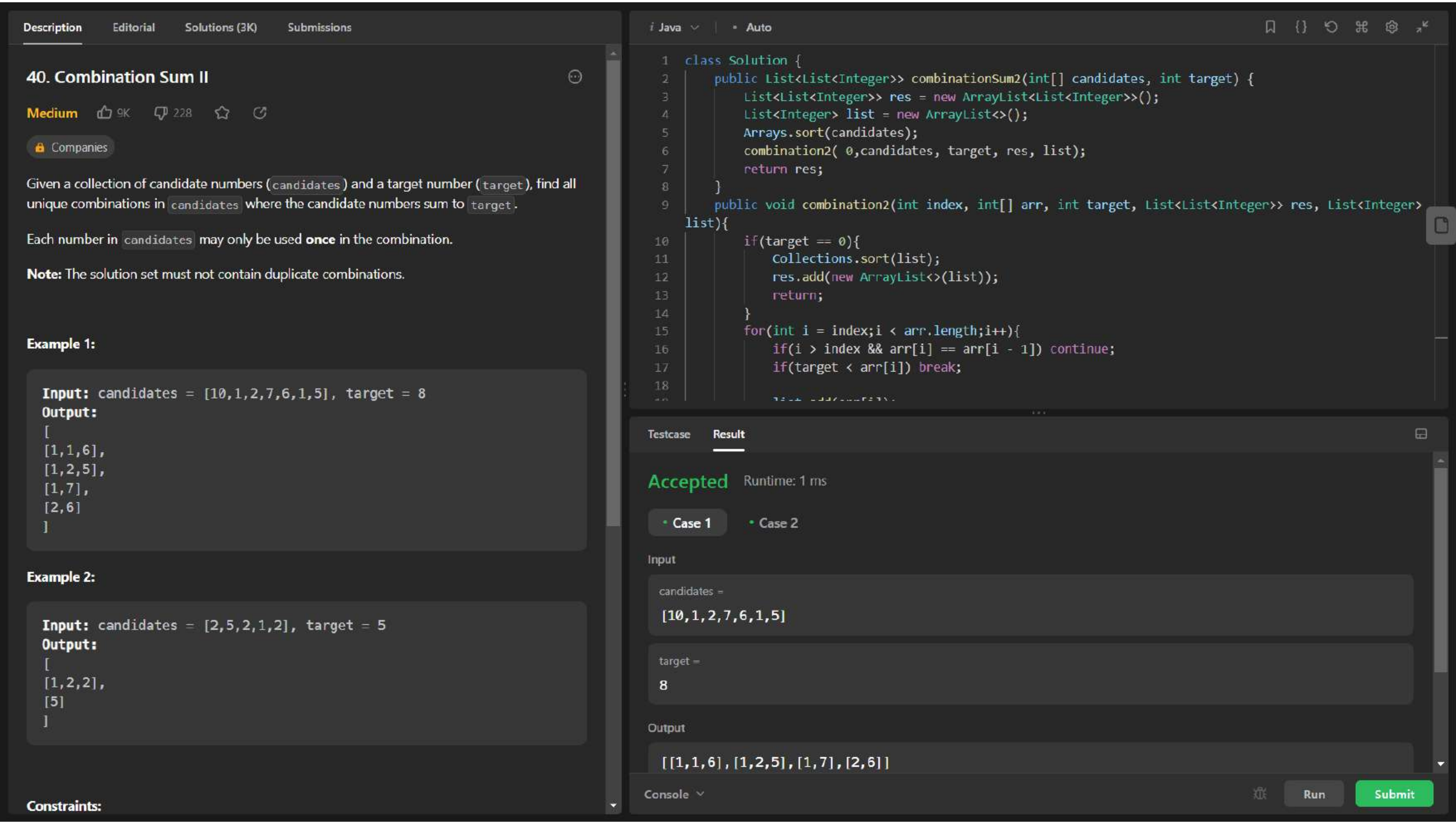


40. Combination Sum II

Medium 9K 228

Companies

Given a collection of candidate numbers (`candidates`) and a target number (`target`), find all unique combinations in `candidates` where the candidate numbers sum to `target`.

Each number in `candidates` may only be used **once** in the combination.

Note: The solution set must not contain duplicate combinations.

Example 1:

```
Input: candidates = [10,1,2,7,6,1,5], target = 8
Output:
[
[1,1,6],
[1,2,5],
[1,7],
[2,6]
]
```

Example 2:

```
Input: candidates = [2,5,2,1,2], target = 5
Output:
[
[1,2,2],
[5]
]
```

Constraints:

```java
class Solution {
    public List<List<Integer>> combinationSum2(int[] candidates, int target) {
        List<List<Integer>> res = new ArrayList<List<Integer>>();
        List<Integer> list = new ArrayList<>();
        Arrays.sort(candidates);
        combination2( 0,candidates, target, res, list);
        return res;
    }
    public void combination2(int index, int[] arr, int target, List<List<Integer>> res, List<Integer>
list){
        if(target == 0){
            Collections.sort(list);
            res.add(new ArrayList<>(list));
            return;
        }
        for(int i = index;i < arr.length;i++){
            if(i > index && arr[i] == arr[i - 1]) continue;
            if(target < arr[i]) break;

```

Accepted Runtime: 1 ms

Case 1 Case 2

Input

candidates =

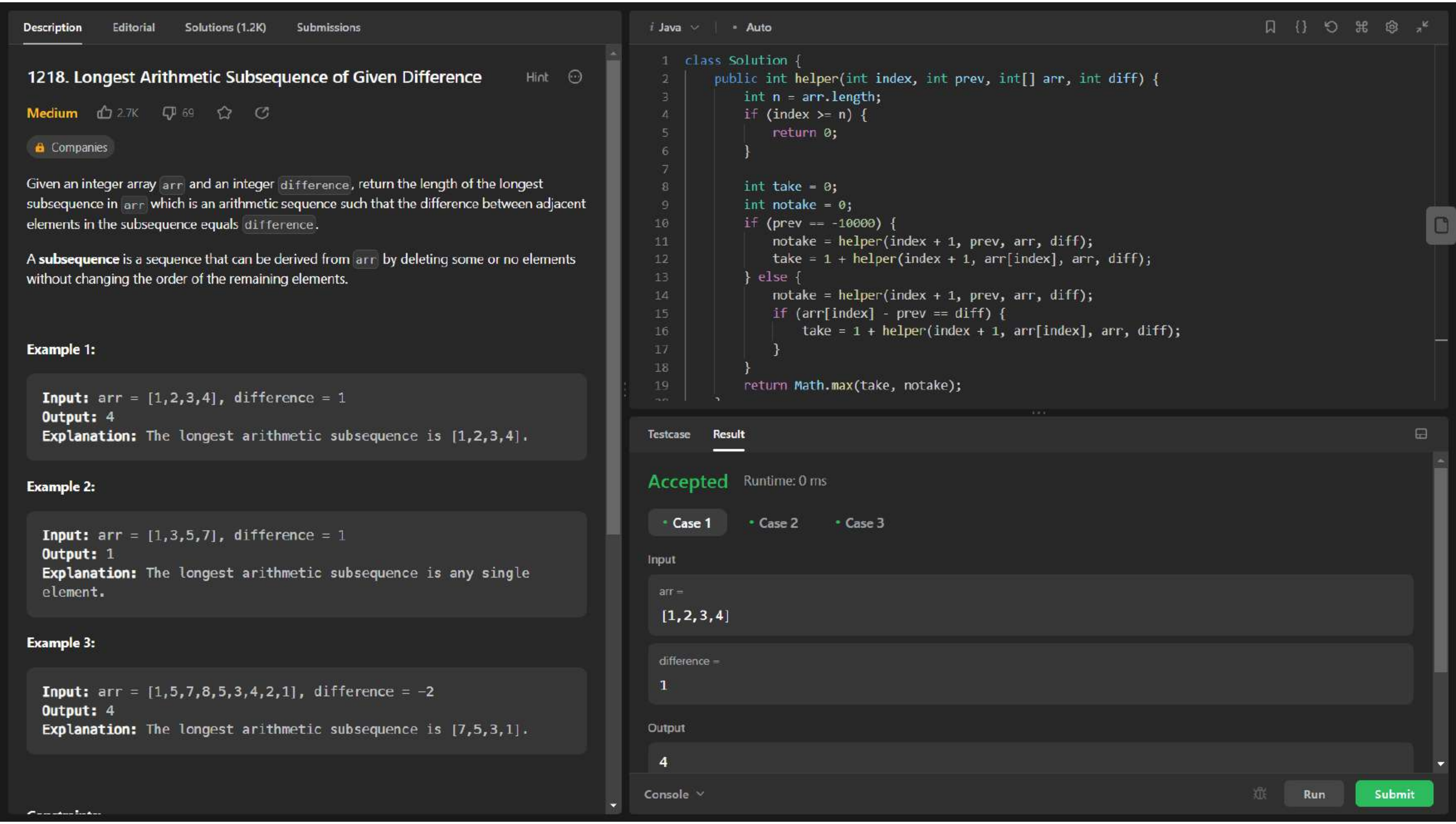
```
[10,1,2,7,6,1,5]
```

target =

```
8
```

Output

```
[[1,1,6],[1,2,5],[1,7],[2,6]]
```



1218. Longest Arithmetic Subsequence of Given Difference

Given an integer array `arr` and an integer `difference`, return the length of the longest subsequence in `arr` which is an arithmetic sequence such that the difference between adjacent elements in the subsequence equals `difference`.

A **subsequence** is a sequence that can be derived from `arr` by deleting some or no elements without changing the order of the remaining elements.

Example 1:

```
Input: arr = [1,2,3,4], difference = 1
Output: 4
Explanation: The longest arithmetic subsequence is [1,2,3,4].
```

Example 2:

```
Input: arr = [1,3,5,7], difference = 1
Output: 1
Explanation: The longest arithmetic subsequence is any single
element.
```

Example 3:

```
Input: arr = [1,5,7,8,5,3,4,2,1], difference = -2
Output: 4
Explanation: The longest arithmetic subsequence is [7,5,3,1].
```

```java
class Solution {
    public int helper(int index, int prev, int[] arr, int diff) {
        int n = arr.length;
        if (index >= n) {
            return 0;
        }

        int take = 0;
        int notake = 0;
        if (prev == -10000) {
            notake = helper(index + 1, prev, arr, diff);
            take = 1 + helper(index + 1, arr[index], arr, diff);
        } else {
            notake = helper(index + 1, prev, arr, diff);
            if (arr[index] - prev == diff) {
                take = 1 + helper(index + 1, arr[index], arr, diff);
            }
        }
        return Math.max(take, notake);
```

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

arr =

```
[1,2,3,4]
```

difference =

```
1
```

Output

```
4
```