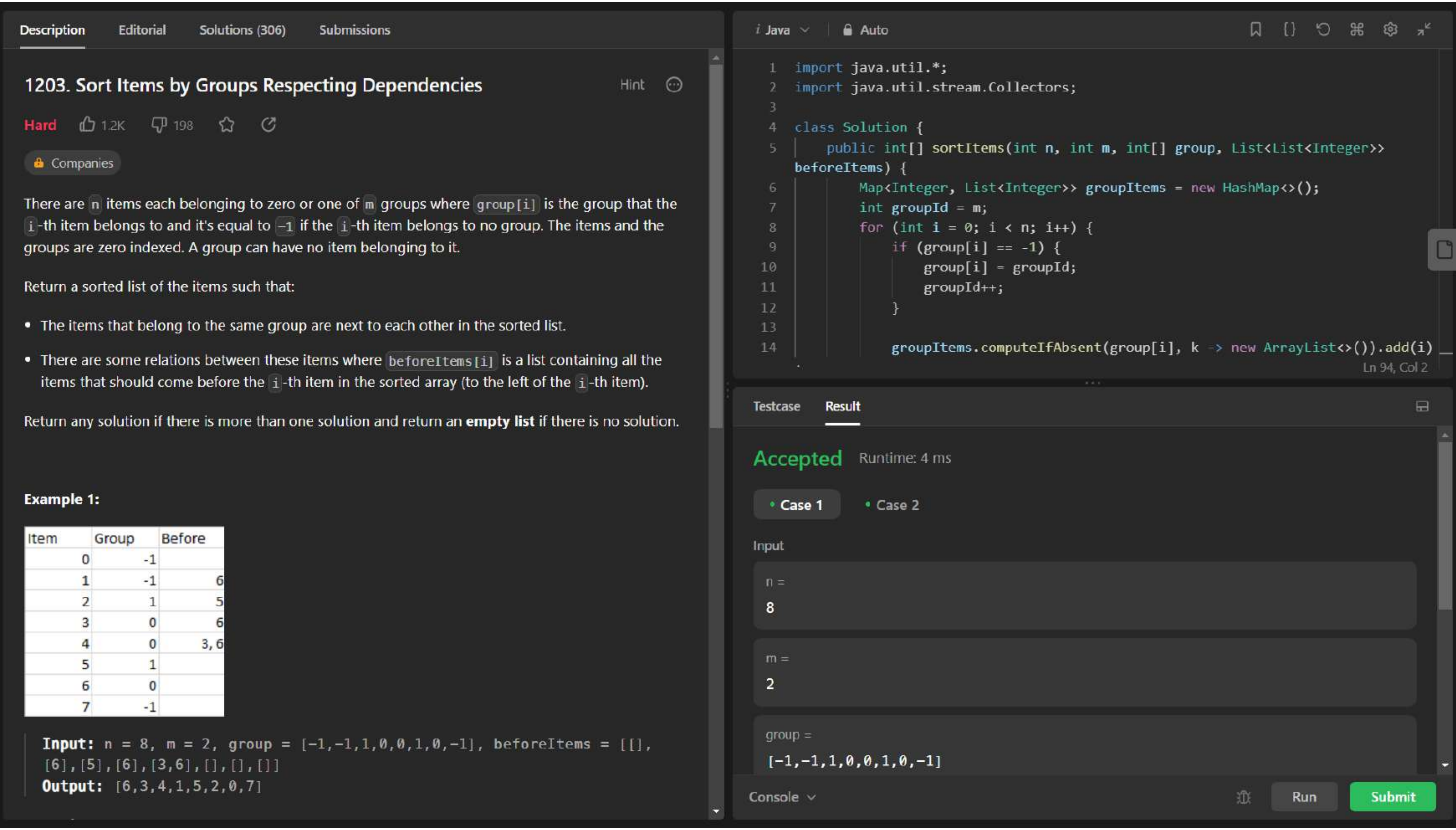


1203. Sort Items by Groups Respecting Dependencies

There are `n` items each belonging to zero or one of `m` groups where `group[i]` is the group that the `i`-th item belongs to and it's equal to `-1` if the `i`-th item belongs to no group. The items and the groups are zero indexed. A group can have no item belonging to it.

Return a sorted list of the items such that:

- The items that belong to the same group are next to each other in the sorted list.
- There are some relations between these items where `beforeItems[i]` is a list containing all the items that should come before the `i`-th item in the sorted array (to the left of the `i`-th item).

Return any solution if there is more than one solution and return an **empty list** if there is no solution.

Example 1:

Item	Group	Before
0	-1	
1	-1	6
2	1	5
3	0	6
4	0	3, 6
5	1	
6	0	
7	-1	

> **Input:** n = 8, m = 2, group = [-1,-1,1,0,0,1,0,-1], beforeItems = [[],[6],[5],[6],[3,6],[],[],[]]
> **Output:** [6,3,4,1,5,2,0,7]

```java
import java.util.*;
import java.util.stream.Collectors;

class Solution {
    public int[] sortItems(int n, int m, int[] group, List<List<Integer>> beforeItems) {
        Map<Integer, List<Integer>> groupItems = new HashMap<>();
        int groupId = m;
        for (int i = 0; i < n; i++) {
            if (group[i] == -1) {
                group[i] = groupId;
                groupId++;
            }

            groupItems.computeIfAbsent(group[i], k -> new ArrayList<>()).add(i)
```

Accepted Runtime: 4 ms

• Case 1 • Case 2

Input

n =

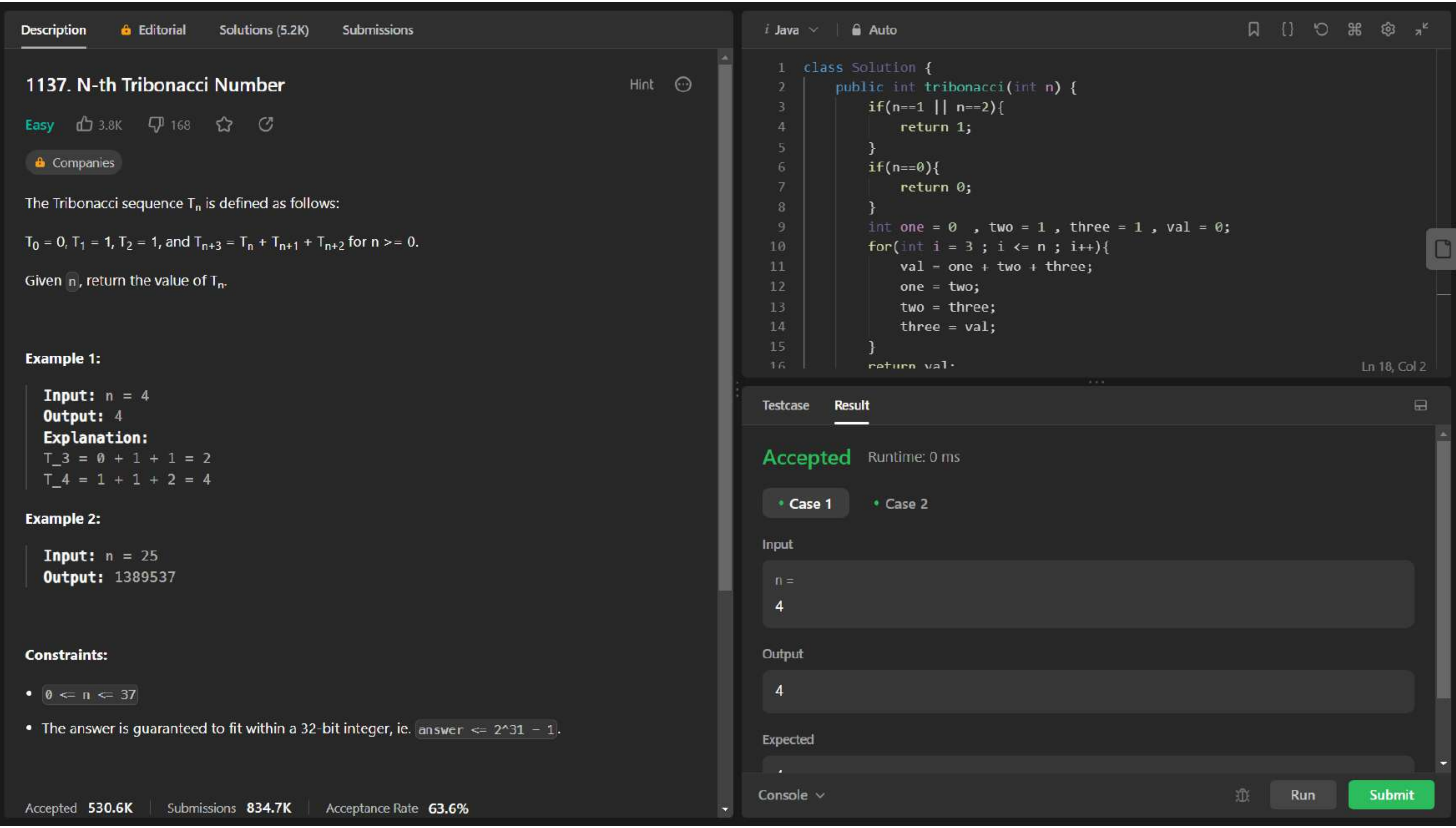
8

m =

2

group =

[-1,-1,1,0,0,1,0,-1]



1137. N-th Tribonacci Number

The Tribonacci sequence T_n is defined as follows:

$T_0 = 0, T_1 = 1, T_2 = 1$, and $T_{n+3} = T_n + T_{n+1} + T_{n+2}$ for n >= 0.

Given `n`, return the value of T_n.

Example 1:

```
Input: n = 4
Output: 4
Explanation:
T_3 = 0 + 1 + 1 = 2
T_4 = 1 + 1 + 2 = 4
```

Example 2:

```
Input: n = 25
Output: 1389537
```

Constraints:

- `0 <= n <= 37`
- The answer is guaranteed to fit within a 32-bit integer, ie. `answer <= 2^31 - 1`.

Accepted 530.6K | Submissions 834.7K | Acceptance Rate 63.6%

```java
class Solution {
    public int tribonacci(int n) {
        if(n==1 || n==2){
            return 1;
        }
        if(n==0){
            return 0;
        }
        int one = 0  , two = 1 , three = 1 , val = 0;
        for(int i = 3 ; i <= n ; i++){
            val = one + two + three;
            one = two;
            two = three;
            three = val;
        }
        return val;
```

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

n =

4

Output

4

Expected