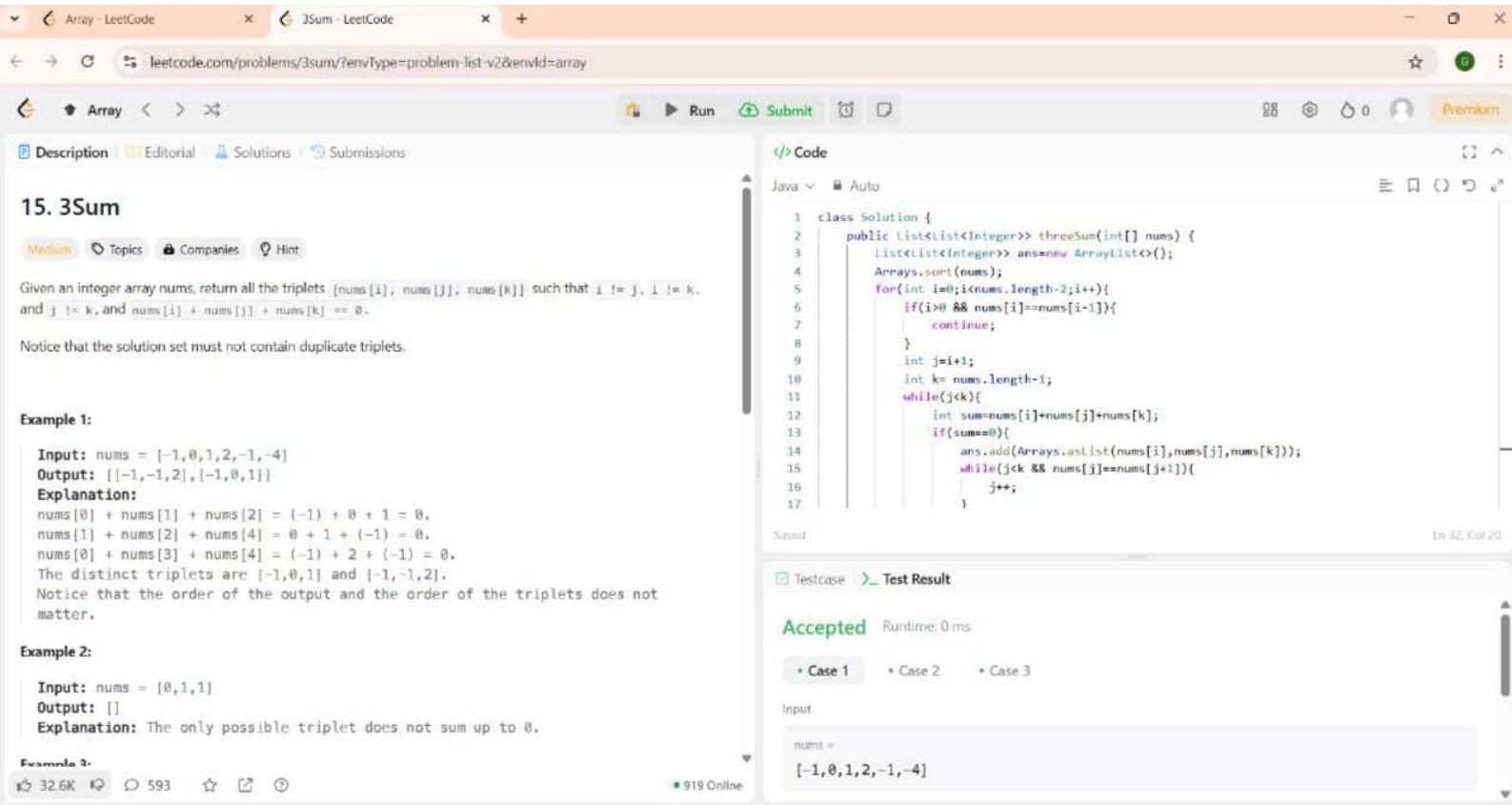
Array - LeetCode
3Sum - LeetCode
leetcode.com/problems/3sum/?envType=problem-list-v2&envId=array
Array
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
15. 3Sum
Medium
Topics
Companies
Hint
Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0.
Notice that the solution set must not contain duplicate triplets.
Example 1:
Input: nums = [-1,0,1,2,-1,-4]
Output: [[-1,-1,2],[-1,0,1]]
Explanation:
nums[0] + nums[1] + nums[2] = (-1) + 0 + 1 = 0.
nums[1] + nums[2] + nums[4] = 0 + 1 + (-1) = 0.
nums[0] + nums[3] + nums[4] = (-1) + 2 + (-1) = 0.
The distinct triplets are [-1,0,1] and [-1,-1,2].
Notice that the order of the output and the order of the triplets does not matter.
Example 2:
Input: nums = [0,1,1]
Output: []
Explanation: The only possible triplet does not sum up to 0.
Example 3:
32.6K
593
919 Online
Code
Java
Auto
class Solution {
public List<List<Integer>> threeSum(int[] nums) {
List<List<Integer>> ans=new ArrayList<>();
Arrays.sort(nums);
for(int i=0;i<nums.length-2;i++){
if(i>0 && nums[i]==nums[i-1]){
continue;
}
int j=i+1;
int k= nums.length-1;
while(j<k){
int sum=nums[i]+nums[j]+nums[k];
if(sum==0){
ans.add(Arrays.asList(nums[i],nums[j],nums[k]));
while(j<k && nums[j]==nums[j+1]){
j++;
}
Saved
Ln 32, Col 20
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
nums =
[-1,0,1,2,-1,-4]