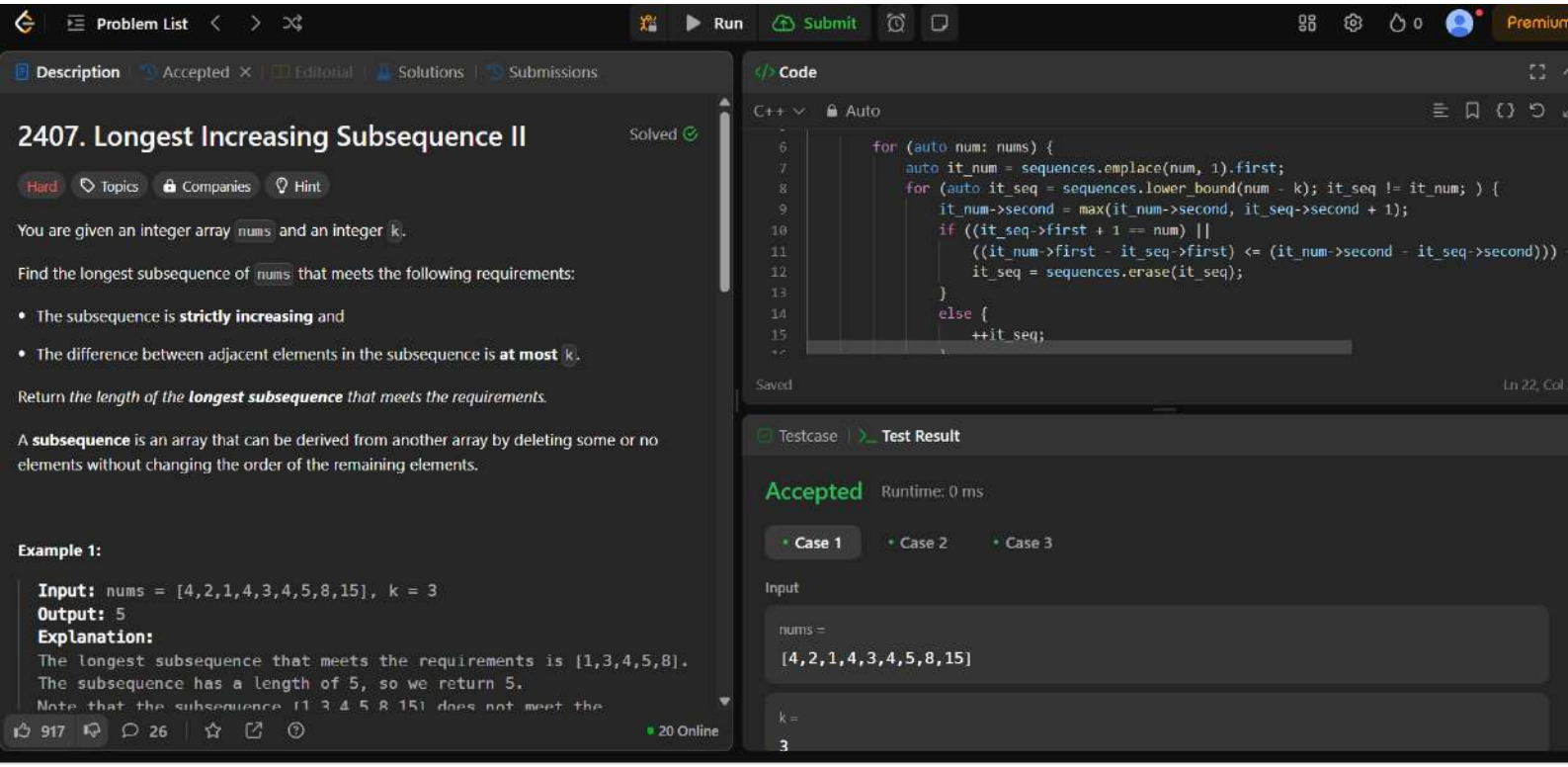
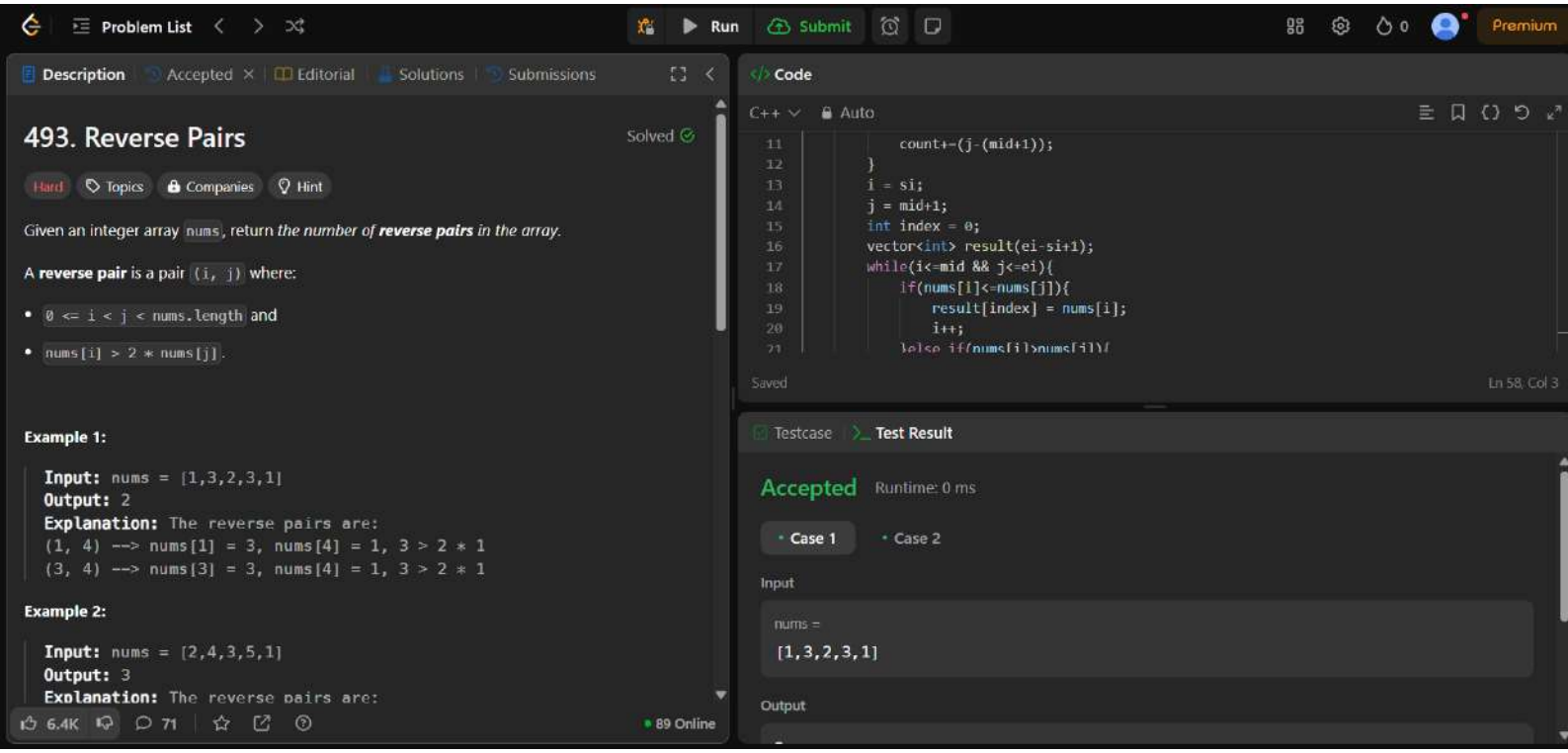
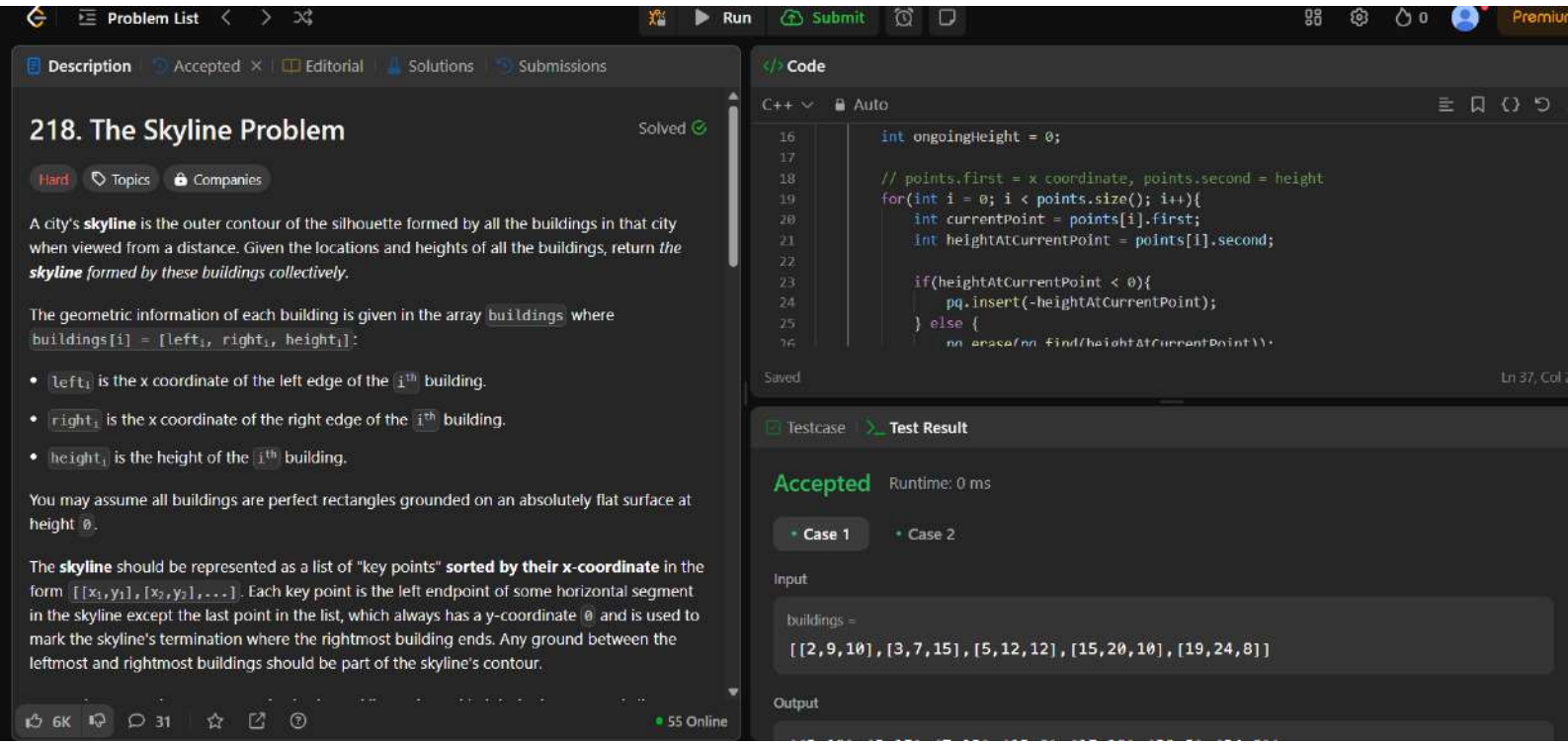
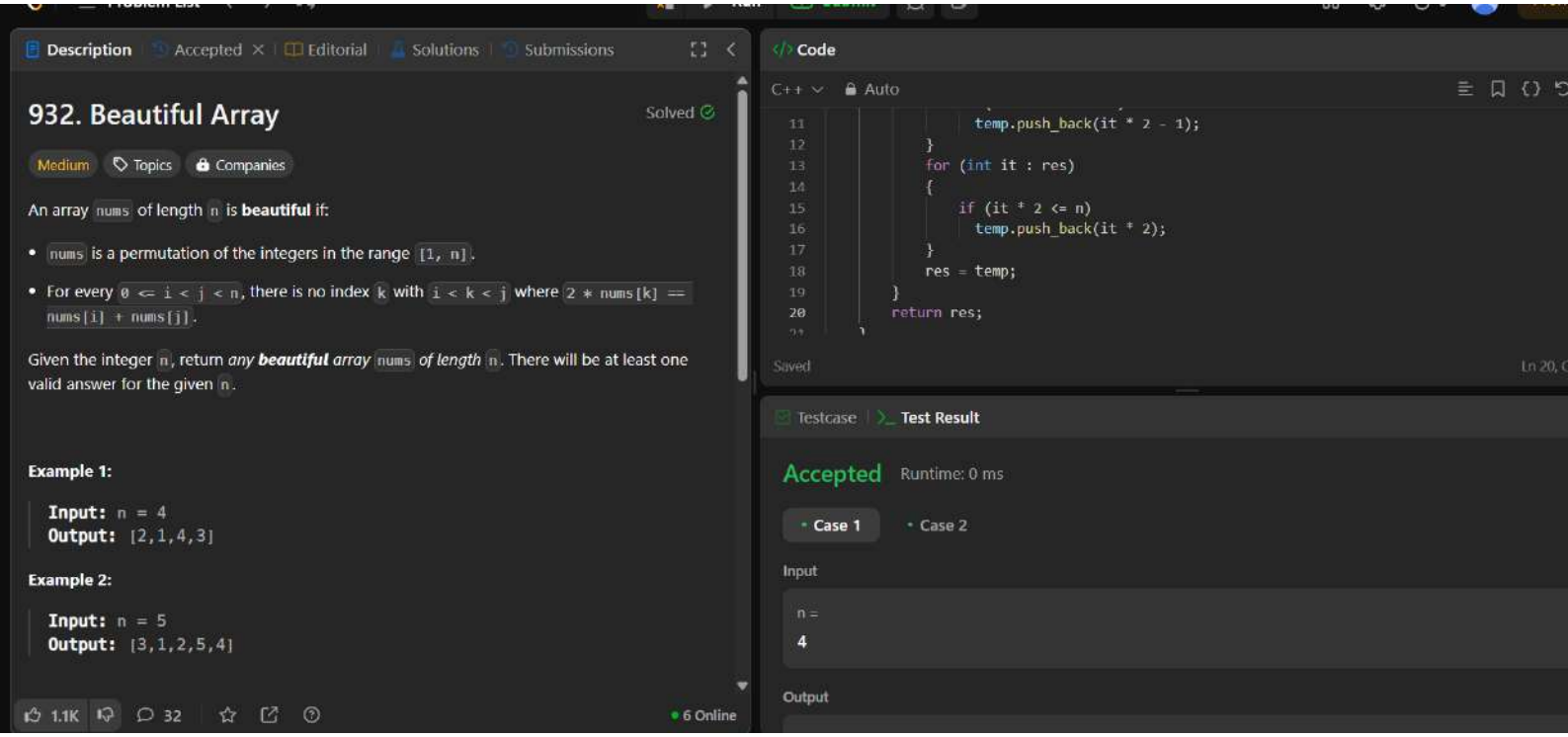
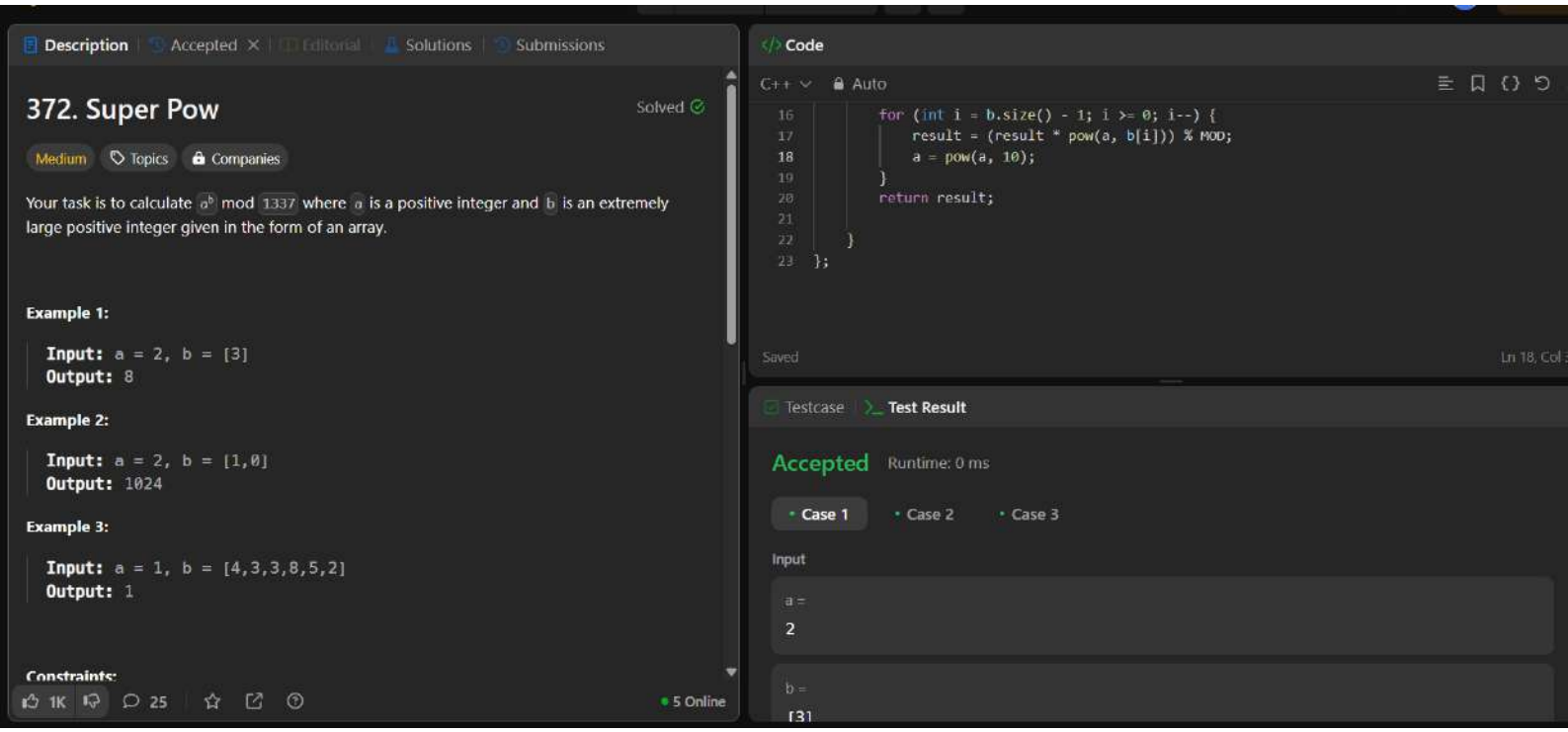

Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
2407. Longest Increasing Subsequence II
Solved
Hard
Topics
Companies
Hint
You are given an integer array nums and an integer k.
Find the longest subsequence of nums that meets the following requirements:
The subsequence is strictly increasing and
The difference between adjacent elements in the subsequence is at most k.
Return the length of the longest subsequence that meets the requirements.
A subsequence is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements.
Example 1:
Input: nums = [4,2,1,4,3,4,5,8,15], k = 3
Output: 5
Explanation:
The longest subsequence that meets the requirements is [1,3,4,5,8].
The subsequence has a length of 5, so we return 5.
Note that the subsequence [1,3,4,5,8,15] does not meet the
917
26
20 Online
Code
C++
Auto
for (auto num: nums) {
auto it_num = sequences.emplace(num, 1).first;
for (auto it_seq = sequences.lower_bound(num - k); it_seq != it_num;) {
it_num->second = max(it_num->second, it_seq->second + 1);
if ((it_seq->first + 1 == num) ||
((it_num->first - it_seq->first) <= (it_num->second - it_seq->second)))
it_seq = sequences.erase(it_seq);
}
else {
++it_seq;
Saved
Ln 22, Col
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
nums =
[4,2,1,4,3,4,5,8,15]
k =
3


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
493. Reverse Pairs
Solved
Hard
Topics
Companies
Hint
Given an integer array nums, return the number of reverse pairs in the array.
A reverse pair is a pair (i, j) where:
0 <= i < j < nums.length and
nums[i] > 2 * nums[j].
Example 1:
Input: nums = [1,3,2,3,1]
Output: 2
Explanation: The reverse pairs are:
(1, 4) --> nums[1] = 3, nums[4] = 1, 3 > 2 * 1
(3, 4) --> nums[3] = 3, nums[4] = 1, 3 > 2 * 1
Example 2:
Input: nums = [2,4,3,5,1]
Output: 3
Explanation: The reverse pairs are:
6.4K
71
89 Online
Code
C++
Auto
count+=(j-(mid+1));
}
i = si;
j = mid+1;
int index = 0;
vector<int> result(ei-si+1);
while(i<=mid && j<=ei){
if(nums[i]<=nums[j]){
result[index] = nums[i];
i++;
Saved
Ln 58, Col 3
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
nums =
[1,3,2,3,1]
Output


Problem List
Run
Submit
Premiu
Description
Accepted
Editorial
Solutions
Submissions
218. The Skyline Problem
Solved
Hard
Topics
Companies
A city's skyline is the outer contour of the silhouette formed by all the buildings in that city when viewed from a distance. Given the locations and heights of all the buildings, return the skyline formed by these buildings collectively.
The geometric information of each building is given in the array buildings where buildings[i] = [left_i, right_i, height_i]:
left_i is the x coordinate of the left edge of the i^th building.
right_i is the x coordinate of the right edge of the i^th building.
height_i is the height of the i^th building.
You may assume all buildings are perfect rectangles grounded on an absolutely flat surface at height 0.
The skyline should be represented as a list of "key points" sorted by their x-coordinate in the form [[x1,y1],[x2,y2],...]. Each key point is the left endpoint of some horizontal segment in the skyline except the last point in the list, which always has a y-coordinate 0 and is used to mark the skyline's termination where the rightmost building ends. Any ground between the leftmost and rightmost buildings should be part of the skyline's contour.
6K
31
55 Online
Code
C++
Auto
int ongoingHeight = 0;
// points.first = x coordinate, points.second = height
for(int i = 0; i < points.size(); i++){
int currentPoint = points[i].first;
int heightAtCurrentPoint = points[i].second;
if(heightAtCurrentPoint < 0){
pq.insert(-heightAtCurrentPoint);
} else {
Saved
Ln 37, Col
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
buildings =
[[2,9,10],[3,7,15],[5,12,12],[15,20,10],[19,24,8]]
Output


Description
Accepted
Editorial
Solutions
Submissions
932. Beautiful Array
Solved
Medium
Topics
Companies
An array nums of length n is beautiful if:
nums is a permutation of the integers in the range [1, n].
For every 0 <= i < j < n, there is no index k with i < k < j where 2 * nums[k] == nums[i] + nums[j].
Given the integer n, return any beautiful array nums of length n. There will be at least one valid answer for the given n.
Example 1:
Input: n = 4
Output: [2,1,4,3]
Example 2:
Input: n = 5
Output: [3,1,2,5,4]
1.1K
32
6 Online
Code
C++
Auto
temp.push_back(it * 2 - 1);
}
for (int it : res)
{
if (it * 2 <= n)
temp.push_back(it * 2);
}
res = temp;
}
return res;
}
Saved
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
n =
4
Output


Description | Accepted | Editorial | Solutions | Submissions
372. Super Pow
Solved
Medium
Topics
Companies
Your task is to calculate a^b mod 1337 where a is a positive integer and b is an extremely large positive integer given in the form of an array.
Example 1:
Input: a = 2, b = [3]
Output: 8
Example 2:
Input: a = 2, b = [1,0]
Output: 1024
Example 3:
Input: a = 1, b = [4,3,3,8,5,2]
Output: 1
Constraints:
1K
25
5 Online
Code
C++
Auto
for (int i = b.size() - 1; i >= 0; i--) {
result = (result * pow(a, b[i])) % MOD;
a = pow(a, 10);
}
return result;
}
};
Saved
Ln 18, Col
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
a =
2
b =
[31

Problem List

Run Submit

Description | Accepted | Editorial | Solutions | Submissions

240. Search a 2D Matrix II

Solved

Medium | Topics | Companies

Write an efficient algorithm that searches for a value `target` in an `m x n` integer matrix `matrix`. This matrix has the following properties:

- Integers in each row are sorted in ascending from left to right.
- Integers in each column are sorted in ascending from top to bottom.

Example 1:

1	4	7	11	15
2	5	8	12	19
3	6	9	16	22
10	13	14	17	24

12.3K 87 55 Online

Code

C++ Auto

```cpp
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        for (int i = 0; i < matrix.size(); i++) {
            for (int j = 0; j < matrix[i].size(); j++) {
                if (matrix[i][j] == target) {
                    return true;
                }
            }
        }
```

Saved Ln 4, Col

Testcase | Test Result

Accepted Runtime: 0 ms

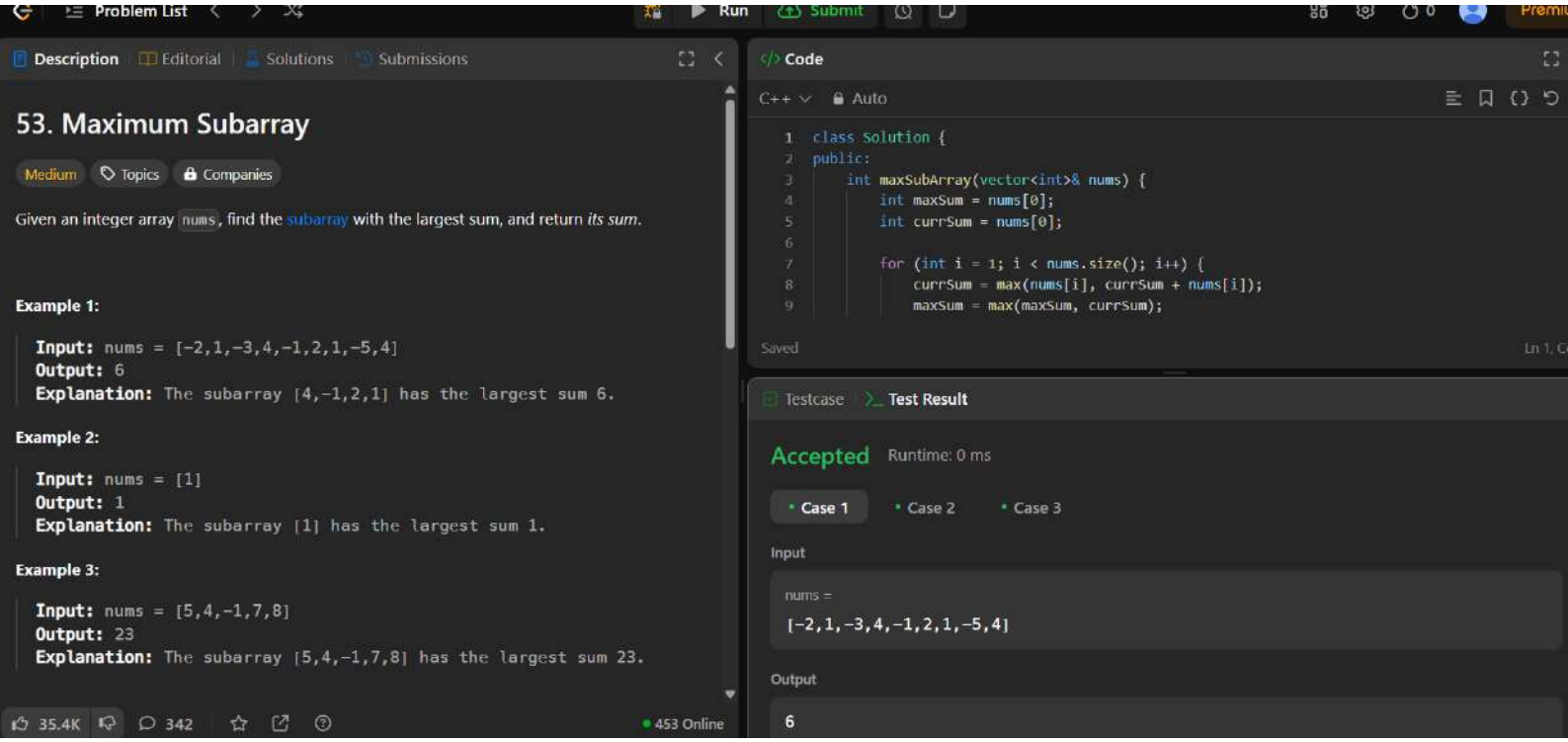
Case 1 | Case 2

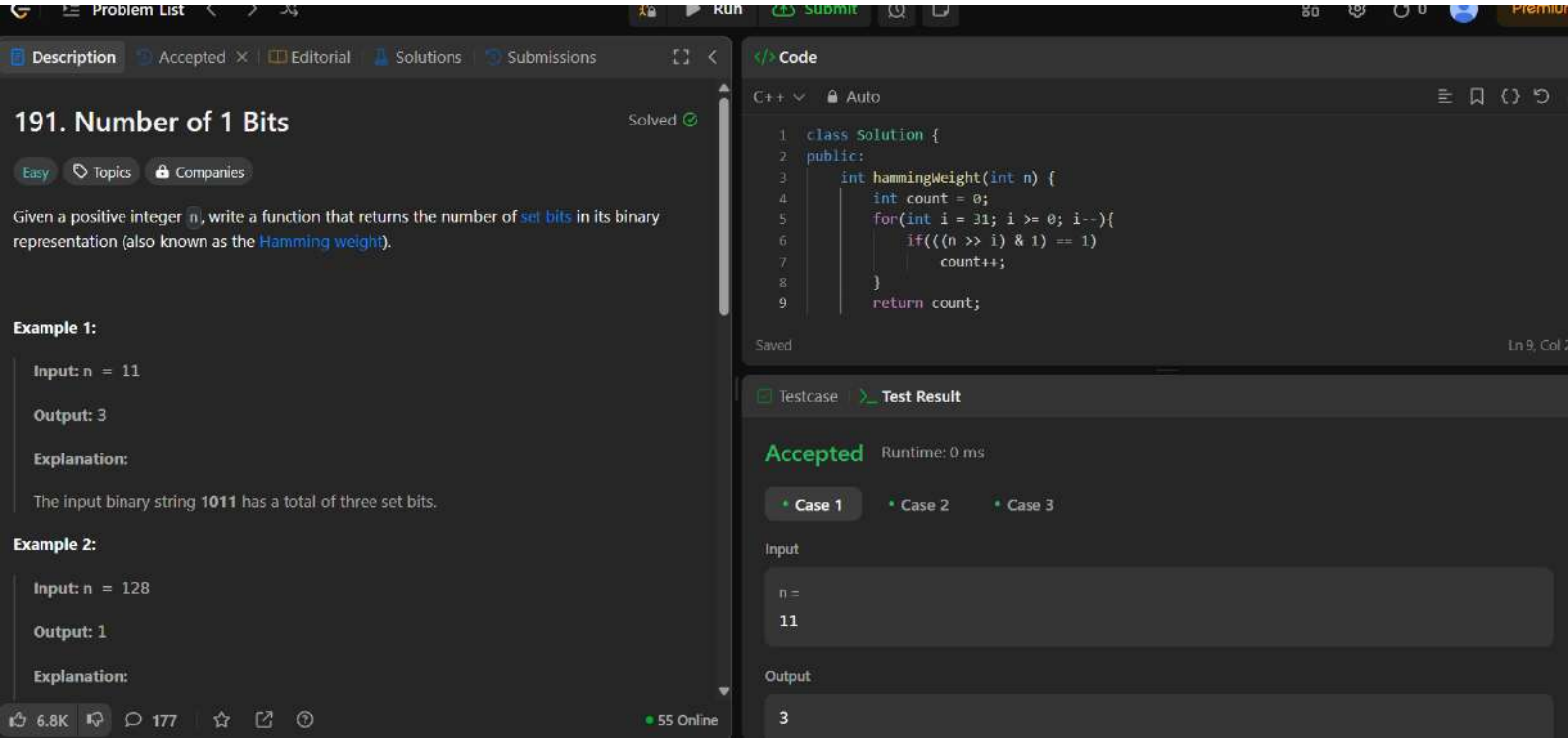
Input

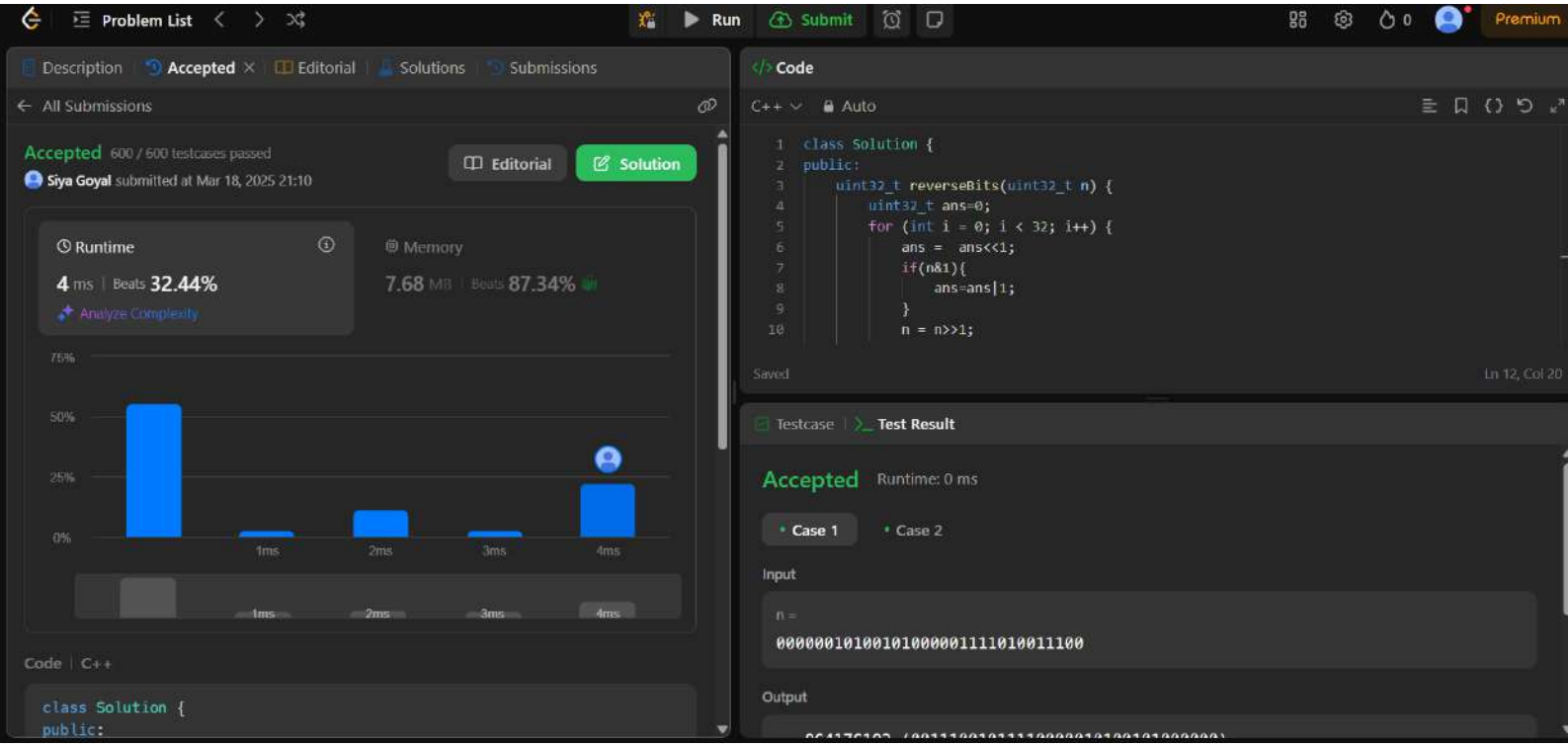
matrix =

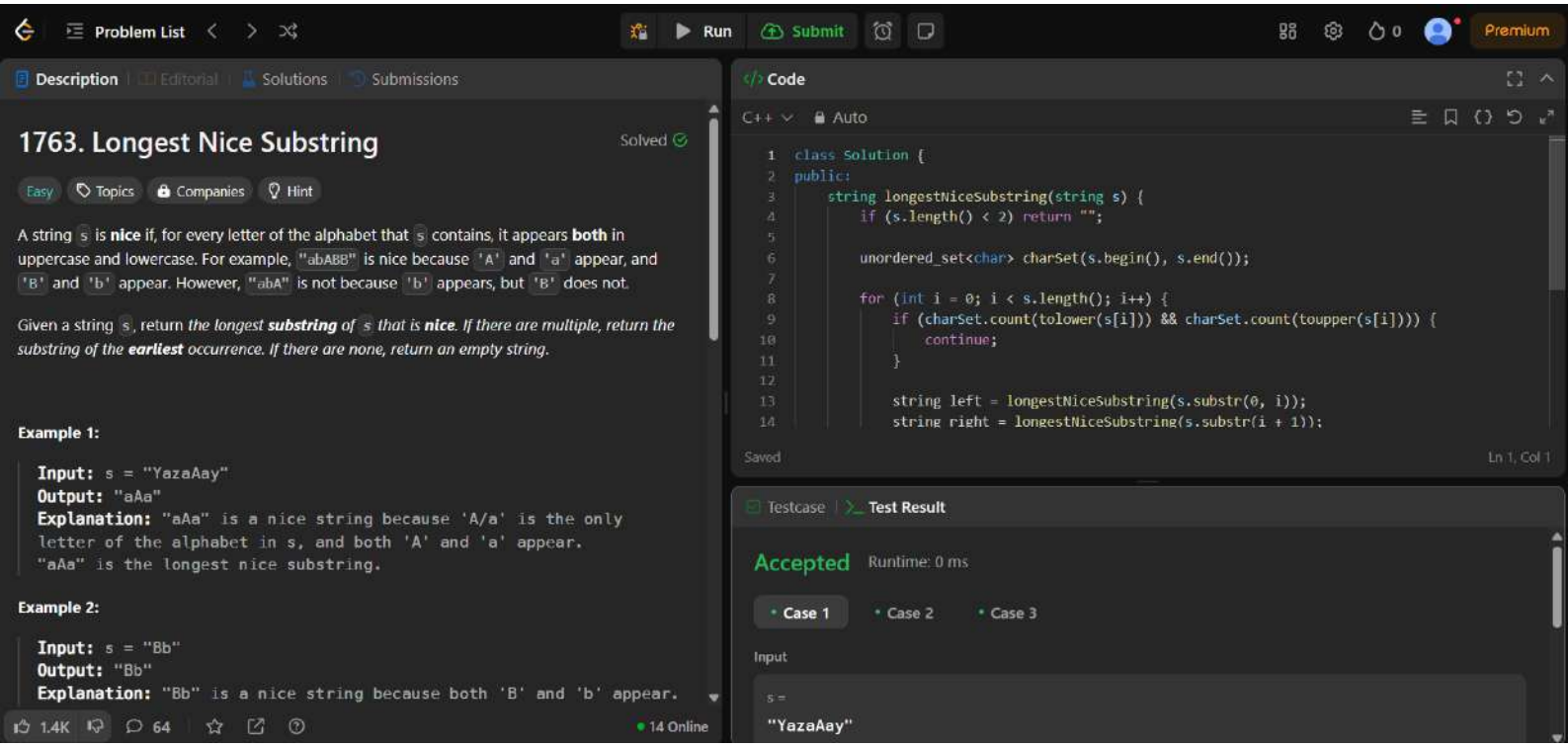
```
[[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]]
```

target =


Problem List
Run
Submit
Description
Editorial
Solutions
Submissions
53. Maximum Subarray
Medium
Topics
Companies
Given an integer array nums, find the subarray with the largest sum, and return its sum.
Example 1:
Input: nums = [-2,1,-3,4,-1,2,1,-5,4]
Output: 6
Explanation: The subarray [4,-1,2,1] has the largest sum 6.
Example 2:
Input: nums = [1]
Output: 1
Explanation: The subarray [1] has the largest sum 1.
Example 3:
Input: nums = [5,4,-1,7,8]
Output: 23
Explanation: The subarray [5,4,-1,7,8] has the largest sum 23.
35.4K
342
453 Online
Code
C++
Auto
class Solution {
public:
int maxSubArray(vector<int>& nums) {
int maxSum = nums[0];
int currSum = nums[0];
for (int i = 1; i < nums.size(); i++) {
currSum = max(nums[i], currSum + nums[i]);
maxSum = max(maxSum, currSum);
Saved
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
nums =
[-2,1,-3,4,-1,2,1,-5,4]
Output
6


Problem List
Run
Submit
Description
Accepted
Editorial
Solutions
Submissions
191. Number of 1 Bits
Solved
Easy
Topics
Companies
Given a positive integer n, write a function that returns the number of set bits in its binary representation (also known as the Hamming weight).
Example 1:
Input: n = 11
Output: 3
Explanation:
The input binary string 1011 has a total of three set bits.
Example 2:
Input: n = 128
Output: 1
Explanation:
6.8K
177
55 Online
Code
C++
Auto
class Solution {
public:
int hammingWeight(int n) {
int count = 0;
for(int i = 31; i >= 0; i--){
if(((n >> i) & 1) == 1)
count++;
}
return count;
Saved
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
n =
11
Output
3


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 600 / 600 testcases passed
Siya Goyal submitted at Mar 18, 2025 21:10
Editorial
Solution
Runtime
4 ms | Beats 32.44%
Analyze Complexity
Memory
7.68 MB | Beats 87.34%
75%
50%
25%
0%
1ms
2ms
3ms
4ms
Code | C++
class Solution {
public:
Code
C++
Auto
class Solution {
public:
uint32_t reverseBits(uint32_t n) {
uint32_t ans=0;
for (int i = 0; i < 32; i++) {
ans = ans<<1;
if(n&1){
ans=ans|1;
}
n = n>>1;
Saved
Ln 12, Col 20
Testcase | Test Result
Accepted Runtime: 0 ms
Case 1
Case 2
Input
n =
00000010100101000001111010011100
Output


Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
1763. Longest Nice Substring
Solved
Easy
Topics
Companies
Hint
A string s is nice if, for every letter of the alphabet that s contains, it appears both in uppercase and lowercase. For example, "abABB" is nice because 'A' and 'a' appear, and 'B' and 'b' appear. However, "abA" is not because 'b' appears, but 'B' does not.
Given a string s, return the longest substring of s that is nice. If there are multiple, return the substring of the earliest occurrence. If there are none, return an empty string.
Example 1:
Input: s = "YazaAay"
Output: "aAa"
Explanation: "aAa" is a nice string because 'A/a' is the only letter of the alphabet in s, and both 'A' and 'a' appear.
"aAa" is the longest nice substring.
Example 2:
Input: s = "Bb"
Output: "Bb"
Explanation: "Bb" is a nice string because both 'B' and 'b' appear.
1.4K
64
14 Online
Code
C++
Auto
class Solution {
public:
string longestNiceSubstring(string s) {
if (s.length() < 2) return "";
unordered_set<char> charSet(s.begin(), s.end());
for (int i = 0; i < s.length(); i++) {
if (charSet.count(tolower(s[i])) && charSet.count(toupper(s[i]))) {
continue;
}
string left = longestNiceSubstring(s.substr(0, i));
string right = longestNiceSubstring(s.substr(i + 1));
Saved
Ln 1, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
s =
"YazaAay"