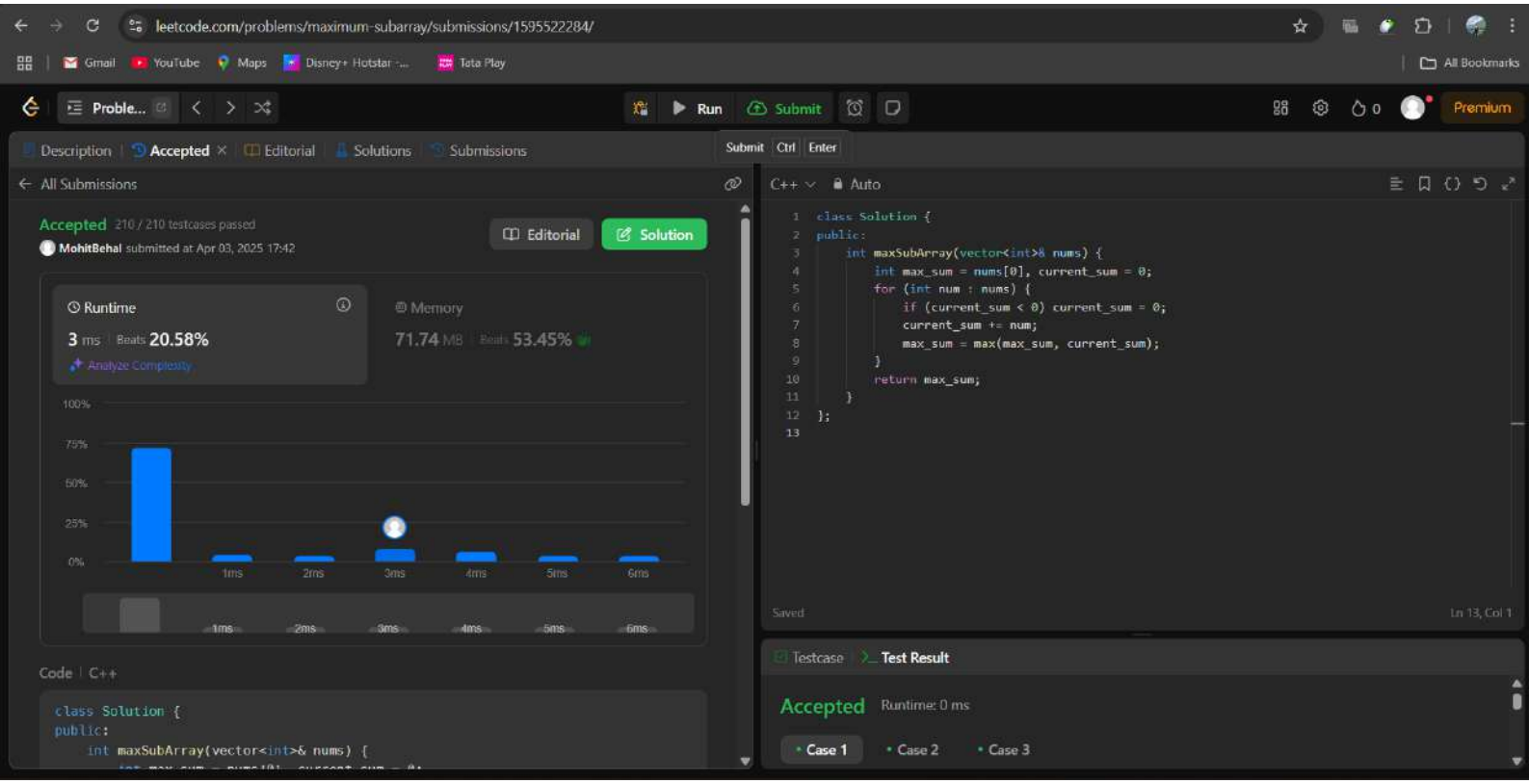
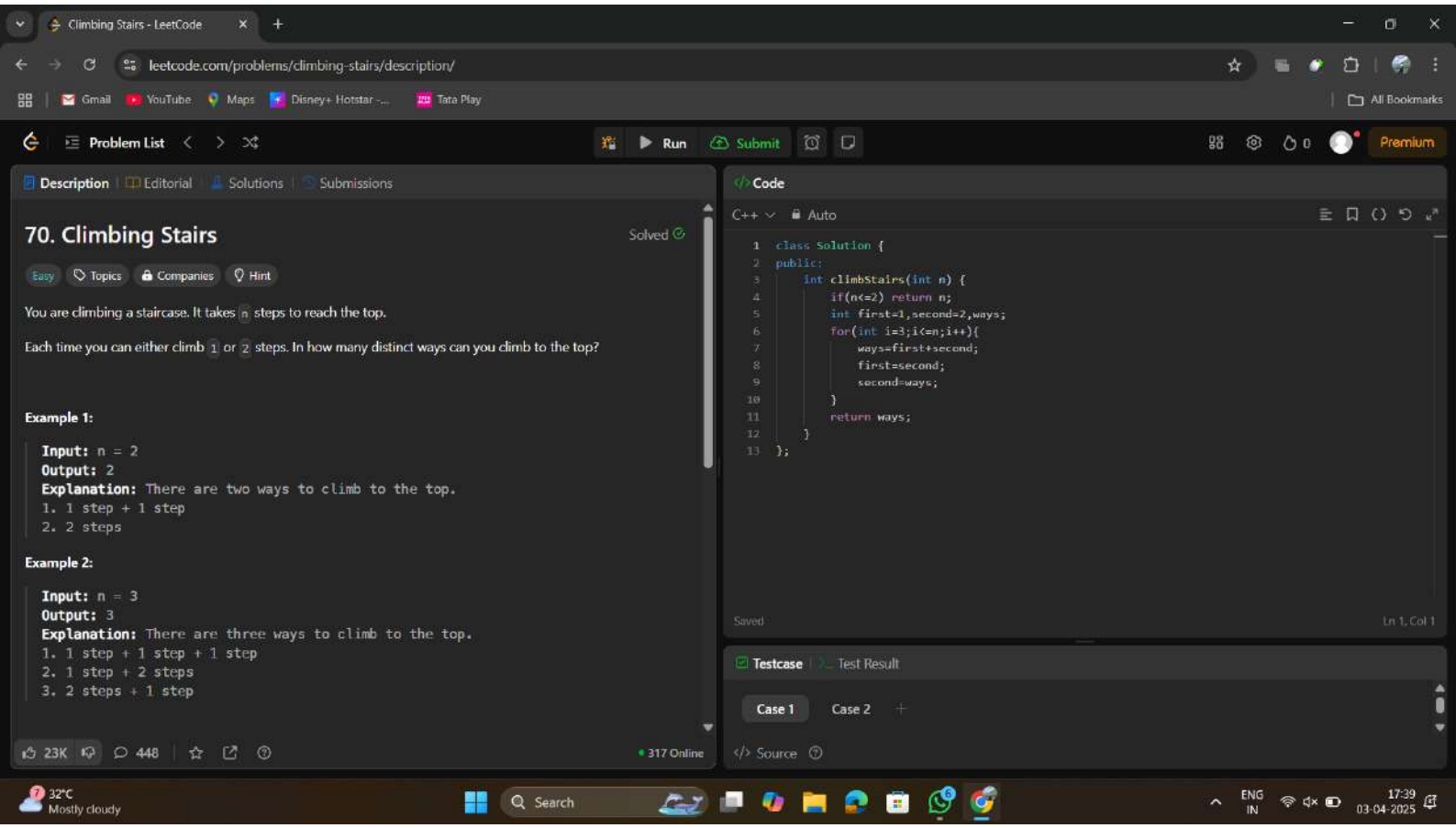
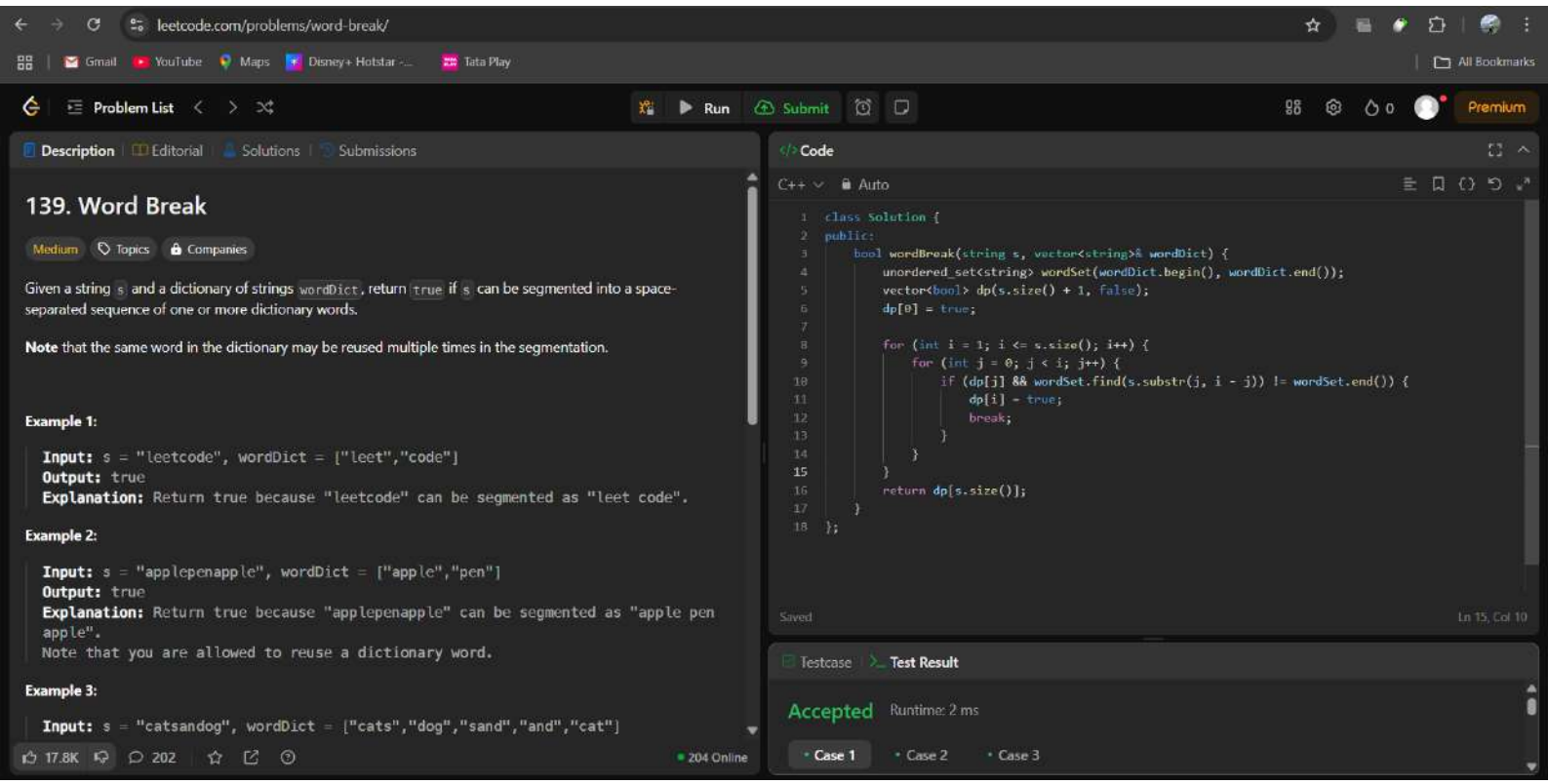
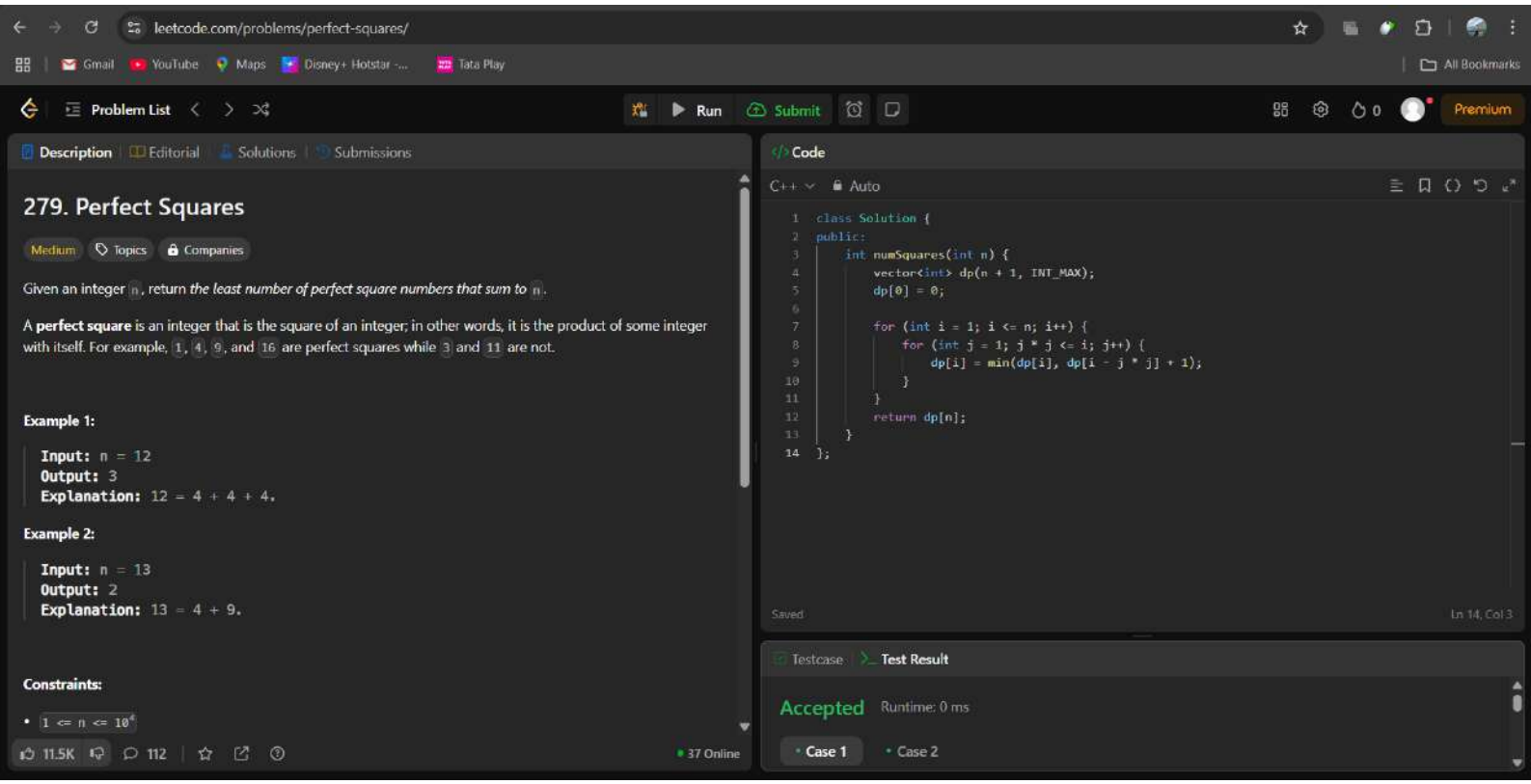
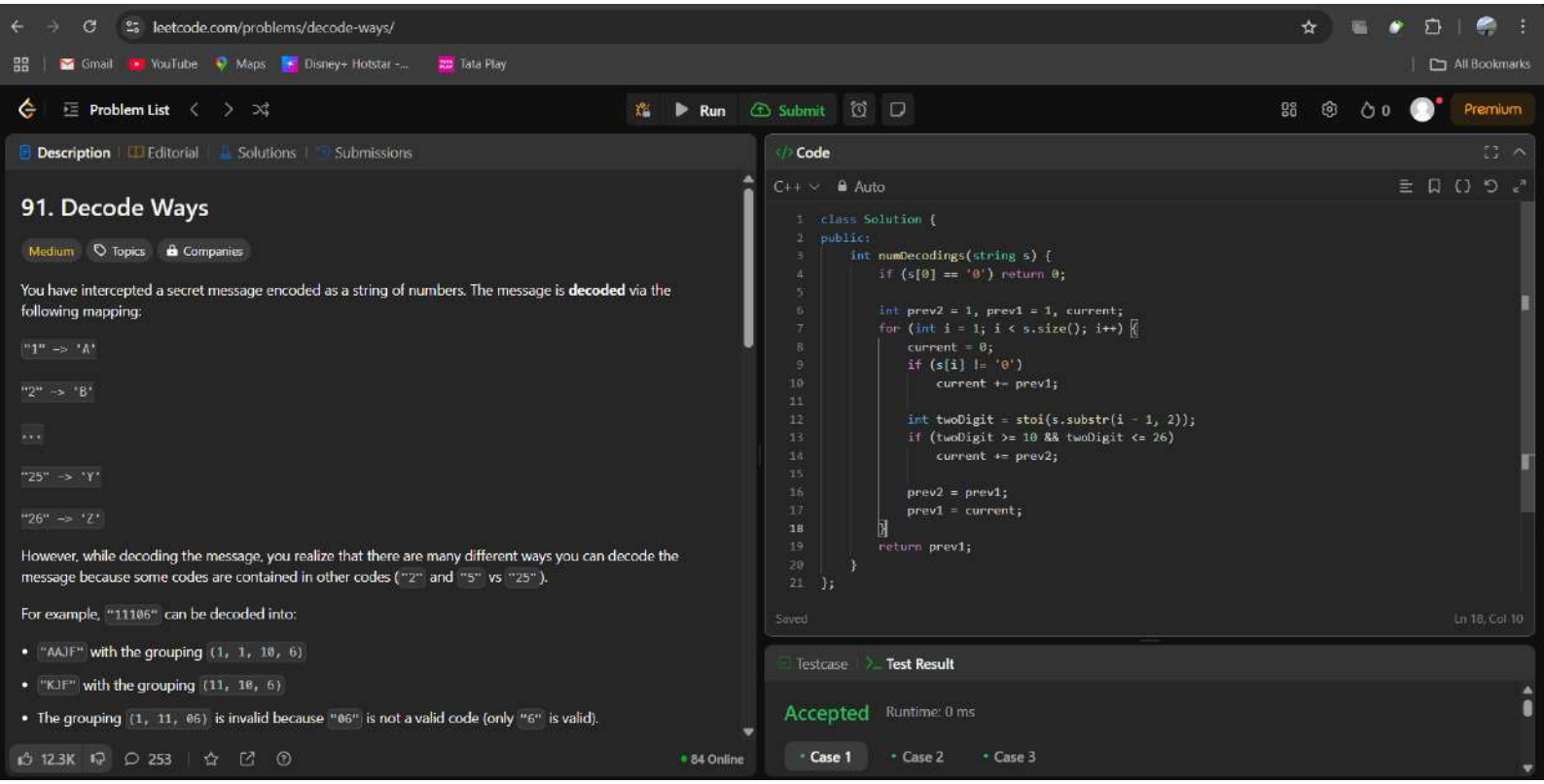
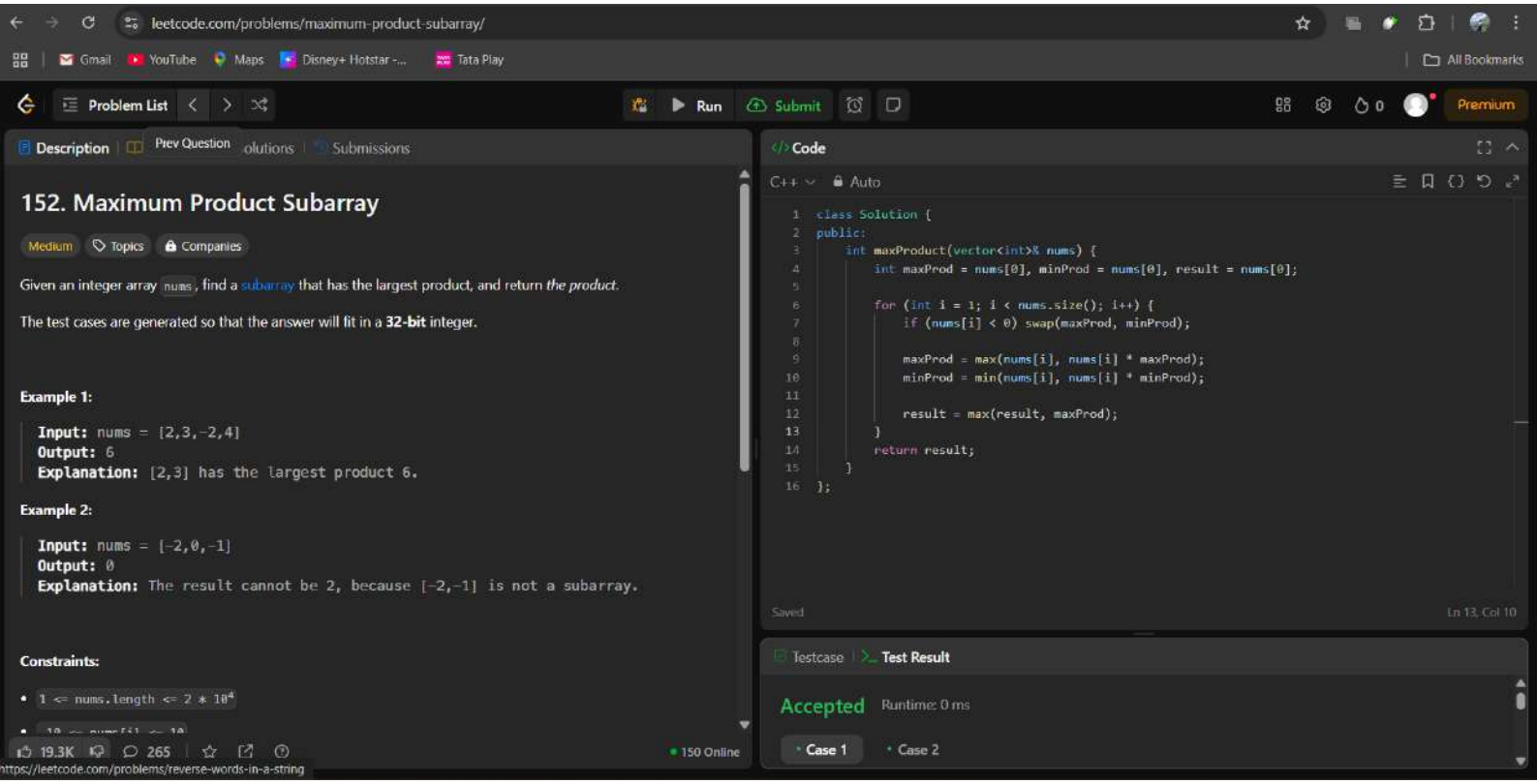

leetcode.com/problems/maximum-subarray/submissions/1595522284/
Gmail
YouTube
Maps
Disney+ Hotstar ...
Tata Play
All Bookmarks
Proble...
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 210 / 210 testcases passed
MohitBehal submitted at Apr 03, 2025 17:42
Editorial
Solution
Runtime
3 ms | Beats 20.58%
Analyze Complexity
Memory
71.74 MB | Beats 53.45%
100%
75%
50%
25%
0%
1ms
2ms
3ms
4ms
5ms
6ms
Code | C++
class Solution {
public:
int maxSubArray(vector<int>& nums) {
Submit Ctrl Enter
C++
Auto
class Solution {
public:
int maxSubArray(vector<int>& nums) {
int max_sum = nums[0], current_sum = 0;
for (int num : nums) {
if (current_sum < 0) current_sum = 0;
current_sum += num;
max_sum = max(max_sum, current_sum);
}
return max_sum;
}
};
Saved
Ln 13, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3

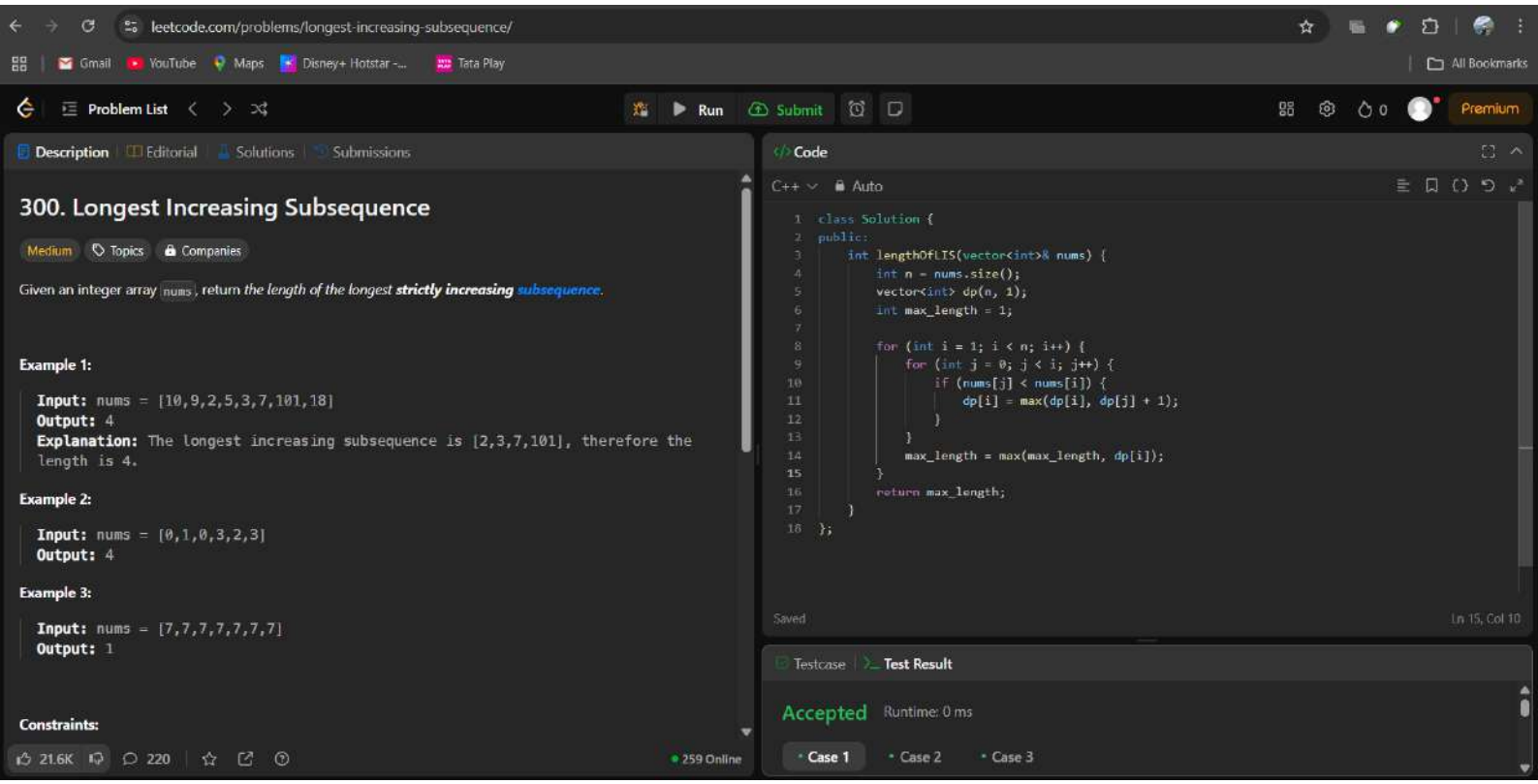

Climbing Stairs - LeetCode
leetcode.com/problems/climbing-stairs/description/
Gmail
YouTube
Maps
Disney+ Hotstar -...
Tata Play
All Bookmarks
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
70. Climbing Stairs
Solved
Easy
Topics
Companies
Hint
You are climbing a staircase. It takes n steps to reach the top.
Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?
Example 1:
Input: n = 2
Output: 2
Explanation: There are two ways to climb to the top.
1. 1 step + 1 step
2. 2 steps
Example 2:
Input: n = 3
Output: 3
Explanation: There are three ways to climb to the top.
1. 1 step + 1 step + 1 step
2. 1 step + 2 steps
3. 2 steps + 1 step
23K
448
317 Online
Code
C++
Auto
class Solution {
public:
int climbStairs(int n) {
if(n<=2) return n;
int first=1,second=2,ways;
for(int i=3;i<=n;i++){
ways=first+second;
first=second;
second=ways;
}
return ways;
}
};
Saved
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Source
32°C
Mostly cloudy
Search
ENG
IN
17:39
03-04-2025

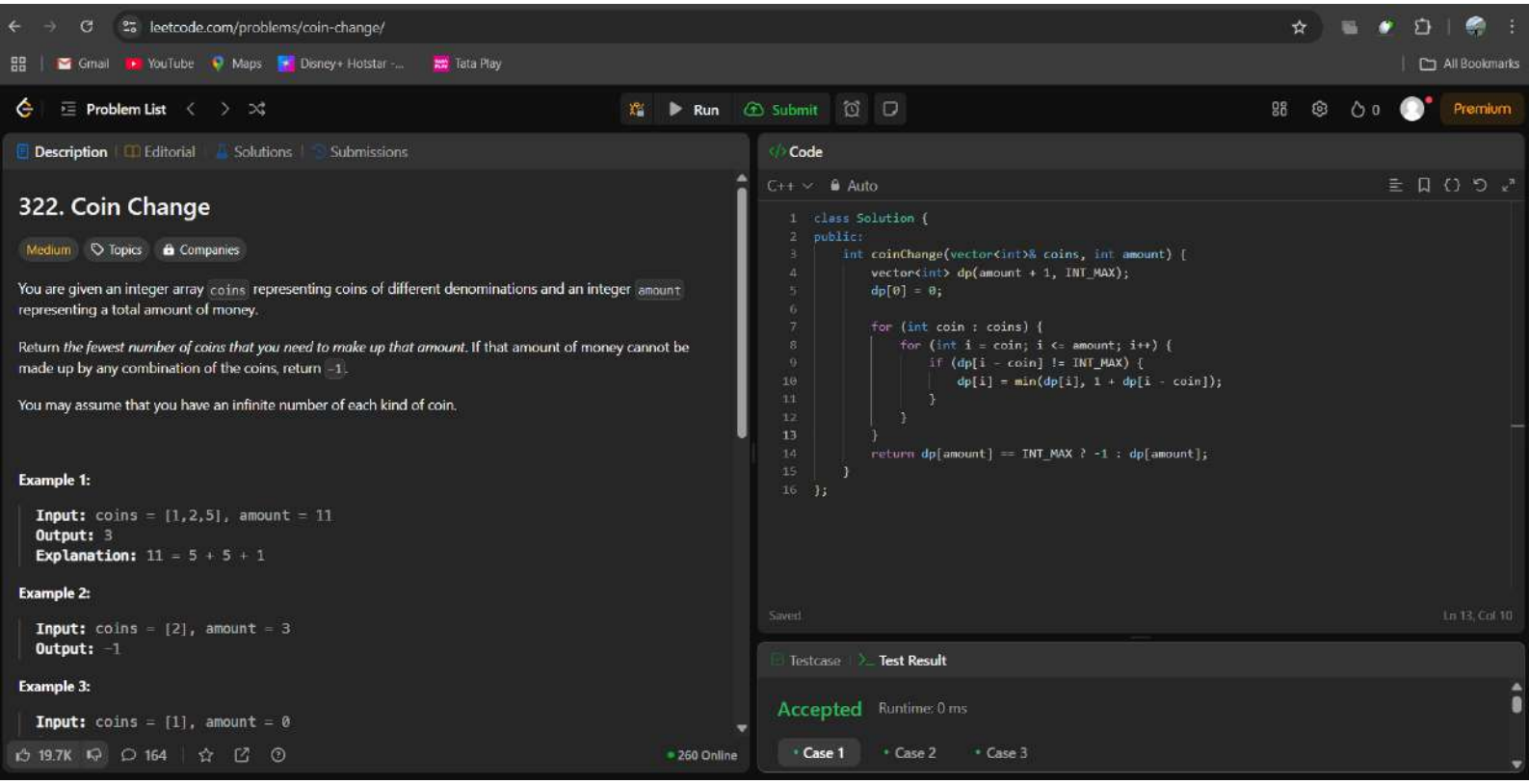

leetcode.com/problems/word-break/
Gmail
YouTube
Maps
Disney+ Hotstar -...
Tata Play
All Bookmarks
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
139. Word Break
Medium
Topics
Companies
Given a string s and a dictionary of strings wordDict, return true if s can be segmented into a space-separated sequence of one or more dictionary words.
Note that the same word in the dictionary may be reused multiple times in the segmentation.
Example 1:
Input: s = "leetcode", wordDict = ["leet","code"]
Output: true
Explanation: Return true because "leetcode" can be segmented as "leet code".
Example 2:
Input: s = "applepenapple", wordDict = ["apple","pen"]
Output: true
Explanation: Return true because "applepenapple" can be segmented as "apple pen apple".
Note that you are allowed to reuse a dictionary word.
Example 3:
Input: s = "catsandog", wordDict = ["cats","dog","sand","and","cat"]
17.8K
202
204 Online
Code
C++
Auto
class Solution {
public:
bool wordBreak(string s, vector<string>& wordDict) {
unordered_set<string> wordSet(wordDict.begin(), wordDict.end());
vector<bool> dp(s.size() + 1, false);
dp[0] = true;
for (int i = 1; i <= s.size(); i++) {
for (int j = 0; j < i; j++) {
if (dp[j] && wordSet.find(s.substr(j, i - j)) != wordSet.end()) {
dp[i] = true;
break;
}
}
}
return dp[s.size()];
}
};
Saved
Ln 15, Col 10
Testcase
Test Result
Accepted
Runtime: 2 ms
Case 1
Case 2
Case 3


leetcode.com/problems/perfect-squares/
Gmail
YouTube
Maps
Disney+ Hotstar -...
Tata Play
All Bookmarks
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
279. Perfect Squares
Medium
Topics
Companies
Given an integer n, return the least number of perfect square numbers that sum to n.
A perfect square is an integer that is the square of an integer; in other words, it is the product of some integer with itself. For example, 1, 4, 9, and 16 are perfect squares while 3 and 11 are not.
Example 1:
Input: n = 12
Output: 3
Explanation: 12 = 4 + 4 + 4.
Example 2:
Input: n = 13
Output: 2
Explanation: 13 = 4 + 9.
Constraints:
1 <= n <= 10^4
11.5K
112
37 Online
Code
C++
Auto
class Solution {
public:
int numSquares(int n) {
vector<int> dp(n + 1, INT_MAX);
dp[0] = 0;
for (int i = 1; i <= n; i++) {
for (int j = 1; j * j <= i; j++) {
dp[i] = min(dp[i], dp[i - j * j] + 1);
}
}
return dp[n];
}
};
Saved
Ln 14, Col 3
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2


leetcode.com/problems/decode-ways/
Gmail
YouTube
Maps
Disney+ Hotstar -...
Tata Play
All Bookmarks
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
91. Decode Ways
Medium
Topics
Companies
You have intercepted a secret message encoded as a string of numbers. The message is decoded via the following mapping:
"1" -> 'A'
"2" -> 'B'
...
"25" -> 'Y'
"26" -> 'Z'
However, while decoding the message, you realize that there are many different ways you can decode the message because some codes are contained in other codes ("2" and "5" vs "25").
For example, "11106" can be decoded into:
"AAJF" with the grouping (1, 1, 10, 6)
"KJF" with the grouping (11, 10, 6)
The grouping (1, 11, 06) is invalid because "06" is not a valid code (only "6" is valid).
12.3K
253
84 Online
Code
C++
Auto
class Solution {
public:
int numDecodings(string s) {
if (s[0] == '0') return 0;
int prev2 = 1, prev1 = 1, current;
for (int i = 1; i < s.size(); i++) {
current = 0;
if (s[i] != '0')
current += prev1;
int twoDigit = stoi(s.substr(i - 1, 2));
if (twoDigit >= 10 && twoDigit <= 26)
current += prev2;
prev2 = prev1;
prev1 = current;
}
return prev1;
}
};
Saved
Ln 18, Col 10
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3


leetcode.com/problems/maximum-product-subarray/
Gmail
YouTube
Maps
Disney+ Hotstar -...
Tata Play
All Bookmarks
Problem List
Run
Submit
Premium
Description
Prev Question
olutions
Submissions
152. Maximum Product Subarray
Medium
Topics
Companies
Given an integer array nums, find a subarray that has the largest product, and return the product.
The test cases are generated so that the answer will fit in a 32-bit integer.
Example 1:
Input: nums = [2,3,-2,4]
Output: 6
Explanation: [2,3] has the largest product 6.
Example 2:
Input: nums = [-2,0,-1]
Output: 0
Explanation: The result cannot be 2, because [-2,-1] is not a subarray.
Constraints:
1 <= nums.length <= 2 * 10^4
19.3K
265
150 Online
https://leetcode.com/problems/reverse-words-in-a-string
Code
C++
Auto
class Solution {
public:
int maxProduct(vector<int>& nums) {
int maxProd = nums[0], minProd = nums[0], result = nums[0];
for (int i = 1; i < nums.size(); i++) {
if (nums[i] < 0) swap(maxProd, minProd);
maxProd = max(nums[i], nums[i] * maxProd);
minProd = min(nums[i], nums[i] * minProd);
result = max(result, maxProd);
}
return result;
}
};
Saved
Ln 13, Col 10
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2


leetcode.com/problems/longest-increasing-subsequence/
Gmail
YouTube
Maps
Disney+ Hotstar -...
Tata Play
All Bookmarks
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
300. Longest Increasing Subsequence
Medium
Topics
Companies
Given an integer array nums, return the length of the longest strictly increasing subsequence.
Example 1:
Input: nums = [10,9,2,5,3,7,101,18]
Output: 4
Explanation: The longest increasing subsequence is [2,3,7,101], therefore the length is 4.
Example 2:
Input: nums = [0,1,0,3,2,3]
Output: 4
Example 3:
Input: nums = [7,7,7,7,7,7,7]
Output: 1
Constraints:
21.6K
220
259 Online
Code
C++
Auto
class Solution {
public:
int lengthOfLIS(vector<int>& nums) {
int n = nums.size();
vector<int> dp(n, 1);
int max_length = 1;
for (int i = 1; i < n; i++) {
for (int j = 0; j < i; j++) {
if (nums[j] < nums[i]) {
dp[i] = max(dp[i], dp[j] + 1);
}
}
max_length = max(max_length, dp[i]);
}
return max_length;
}
};
Saved
Ln 15, Col 10
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3


leetcode.com/problems/coin-change/
Gmail
YouTube
Maps
Disney+ Hotstar -...
Tata Play
All Bookmarks
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
322. Coin Change
Medium
Topics
Companies
You are given an integer array coins representing coins of different denominations and an integer amount representing a total amount of money.
Return the fewest number of coins that you need to make up that amount. If that amount of money cannot be made up by any combination of the coins, return -1.
You may assume that you have an infinite number of each kind of coin.
Example 1:
Input: coins = [1,2,5], amount = 11
Output: 3
Explanation: 11 = 5 + 5 + 1
Example 2:
Input: coins = [2], amount = 3
Output: -1
Example 3:
Input: coins = [1], amount = 0
19.7K
164
260 Online
Code
C++
Auto
class Solution {
public:
int coinChange(vector<int>& coins, int amount) {
vector<int> dp(amount + 1, INT_MAX);
dp[0] = 0;
for (int coin : coins) {
for (int i = coin; i <= amount; i++) {
if (dp[i - coin] != INT_MAX) {
dp[i] = min(dp[i], 1 + dp[i - coin]);
}
}
}
return dp[amount] == INT_MAX ? -1 : dp[amount];
}
};
Saved
Ln 13, Col 10
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3

leetcode.com/problems/unique-paths/

Gmail | YouTube | Maps | Disney+ Hotstar -... | Tata Play | All Bookmarks

Problem List | Run | Submit | Premium

Description | Editorial | Solutions | Submissions

62. Unique Paths

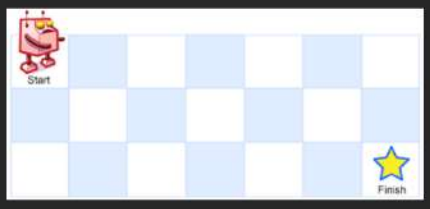
Medium | Topics | Companies

There is a robot on an `m x n` grid. The robot is initially located at the **top-left corner** (i.e., `grid[0][0]`). The robot tries to move to the **bottom-right corner** (i.e., `grid[m - 1][n - 1]`). The robot can only move either down or right at any point in time.

Given the two integers `m` and `n`, return *the number of possible unique paths that the robot can take to reach the bottom-right corner*.

The test cases are generated so that the answer will be less than or equal to `2 * 10^9`.

Example 1:



17.3K | 182 | 132 Online

Code

C++ | Auto

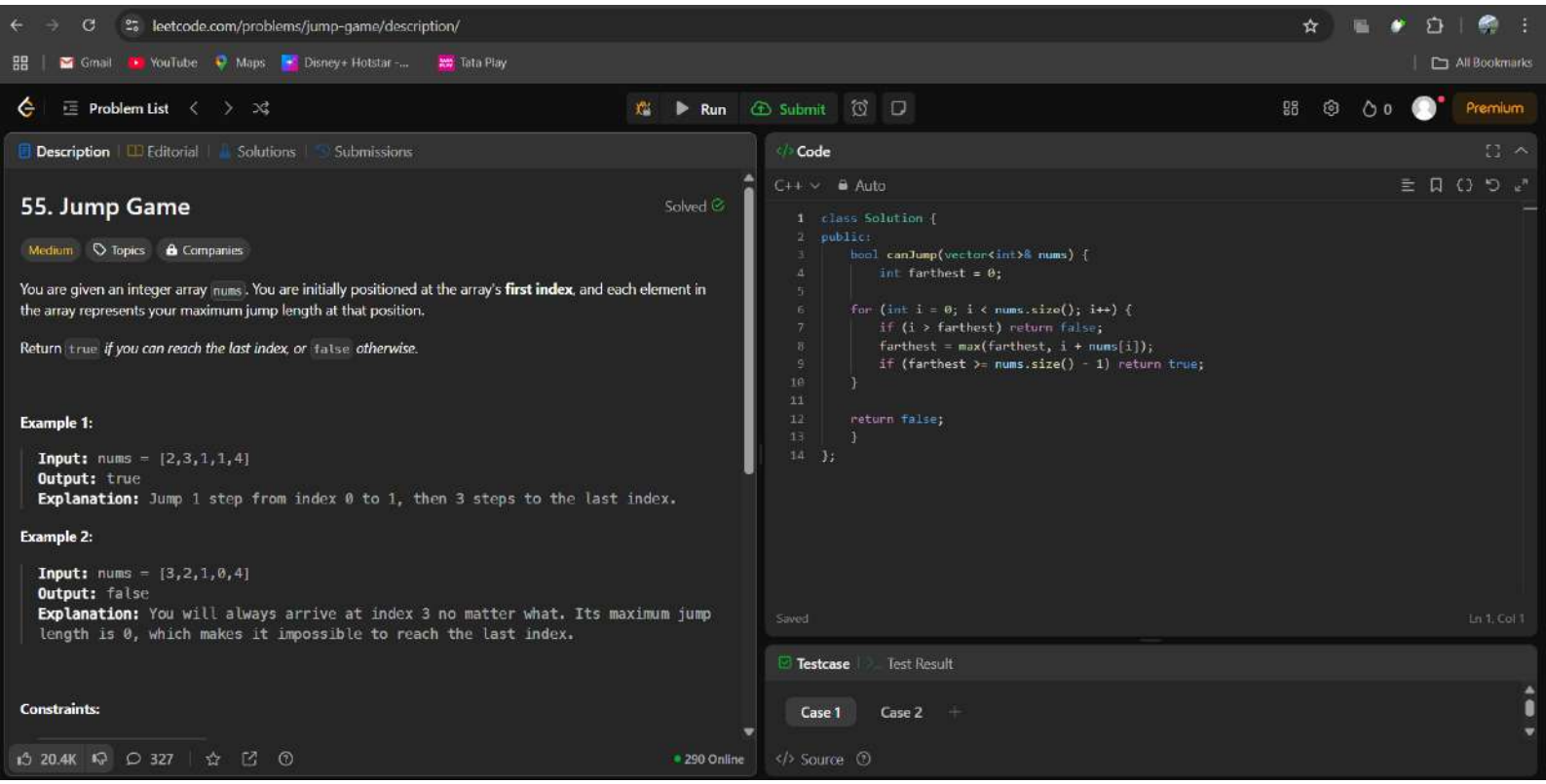
```cpp
class Solution {
public:
    int uniquePaths(int m, int n) {
        vector<vector<int>> dp(m, vector<int>(n, 1));
        for (int i = 1; i < m; i++) {
            for (int j = 1; j < n; j++) {
                dp[i][j] = dp[i - 1][j] + dp[i][j - 1];
            }
        }
        return dp[m - 1][n - 1];
    }
};
```

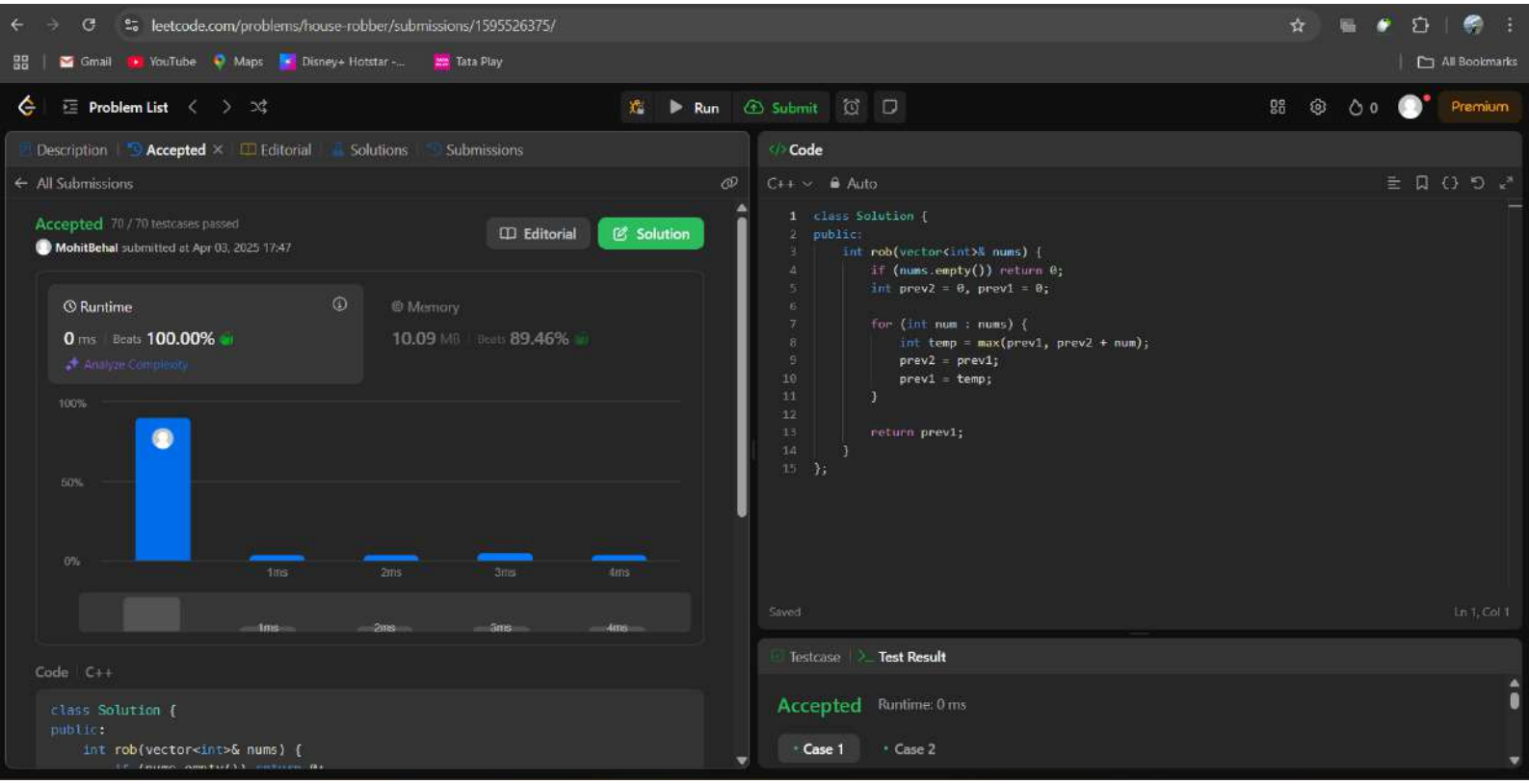
Saved | Ln 13, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2


leetcode.com/problems/jump-game/description/
Gmail
YouTube
Maps
Disney+ Hotstar -...
Tata Play
All Bookmarks
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
55. Jump Game
Solved
Medium
Topics
Companies
You are given an integer array nums. You are initially positioned at the array's first index, and each element in the array represents your maximum jump length at that position.
Return true if you can reach the last index, or false otherwise.
Example 1:
Input: nums = [2,3,1,1,4]
Output: true
Explanation: Jump 1 step from index 0 to 1, then 3 steps to the last index.
Example 2:
Input: nums = [3,2,1,0,4]
Output: false
Explanation: You will always arrive at index 3 no matter what. Its maximum jump length is 0, which makes it impossible to reach the last index.
Constraints:
20.4K
327
290 Online
Code
C++
Auto
class Solution {
public:
bool canJump(vector<int>& nums) {
int farthest = 0;
for (int i = 0; i < nums.size(); i++) {
if (i > farthest) return false;
farthest = max(farthest, i + nums[i]);
if (farthest >= nums.size() - 1) return true;
}
return false;
}
};
Saved
Ln 1, Col 1
Testcase
Test Result
Case 1
Case 2
Source


leetcode.com/problems/house-robber/submissions/1595526375/
Gmail
YouTube
Maps
Disney+ Hotstar -...
Tata Play
All Bookmarks
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 70 / 70 testcases passed
MohitBehal submitted at Apr 03, 2025 17:47
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
10.09 MB | Beats 89.46%
100%
50%
0%
1ms
2ms
3ms
4ms
Code | C++
class Solution {
public:
int rob(vector<int>& nums) {
Code
C++
Auto
1 class Solution {
2 public:
3 int rob(vector<int>& nums) {
4 if (nums.empty()) return 0;
5 int prev2 = 0, prev1 = 0;
6
7 for (int num : nums) {
8 int temp = max(prev1, prev2 + num);
9 prev2 = prev1;
10 prev1 = temp;
11 }
12
13 return prev1;
14 }
15 };
Saved
Ln 1, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2