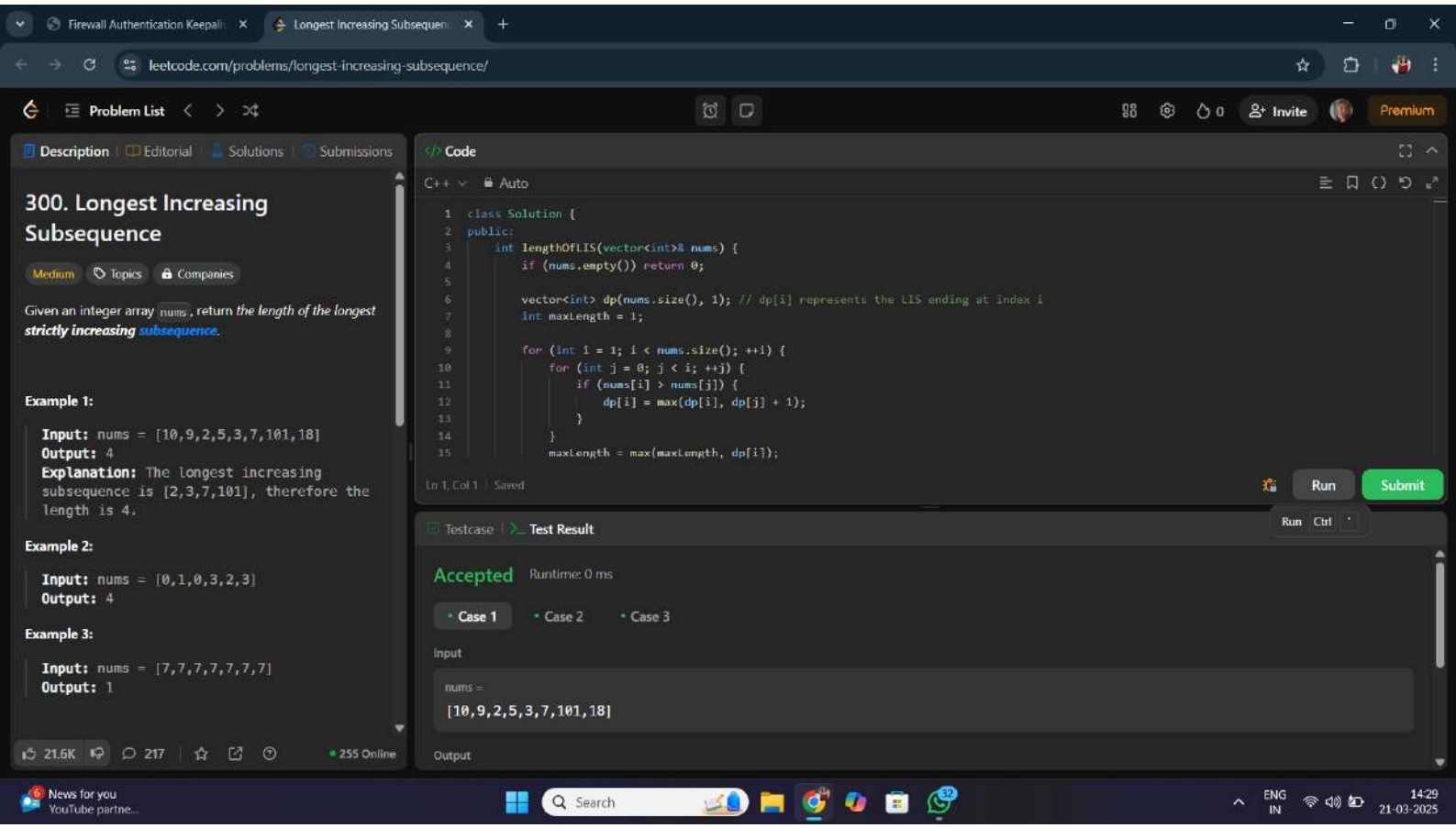
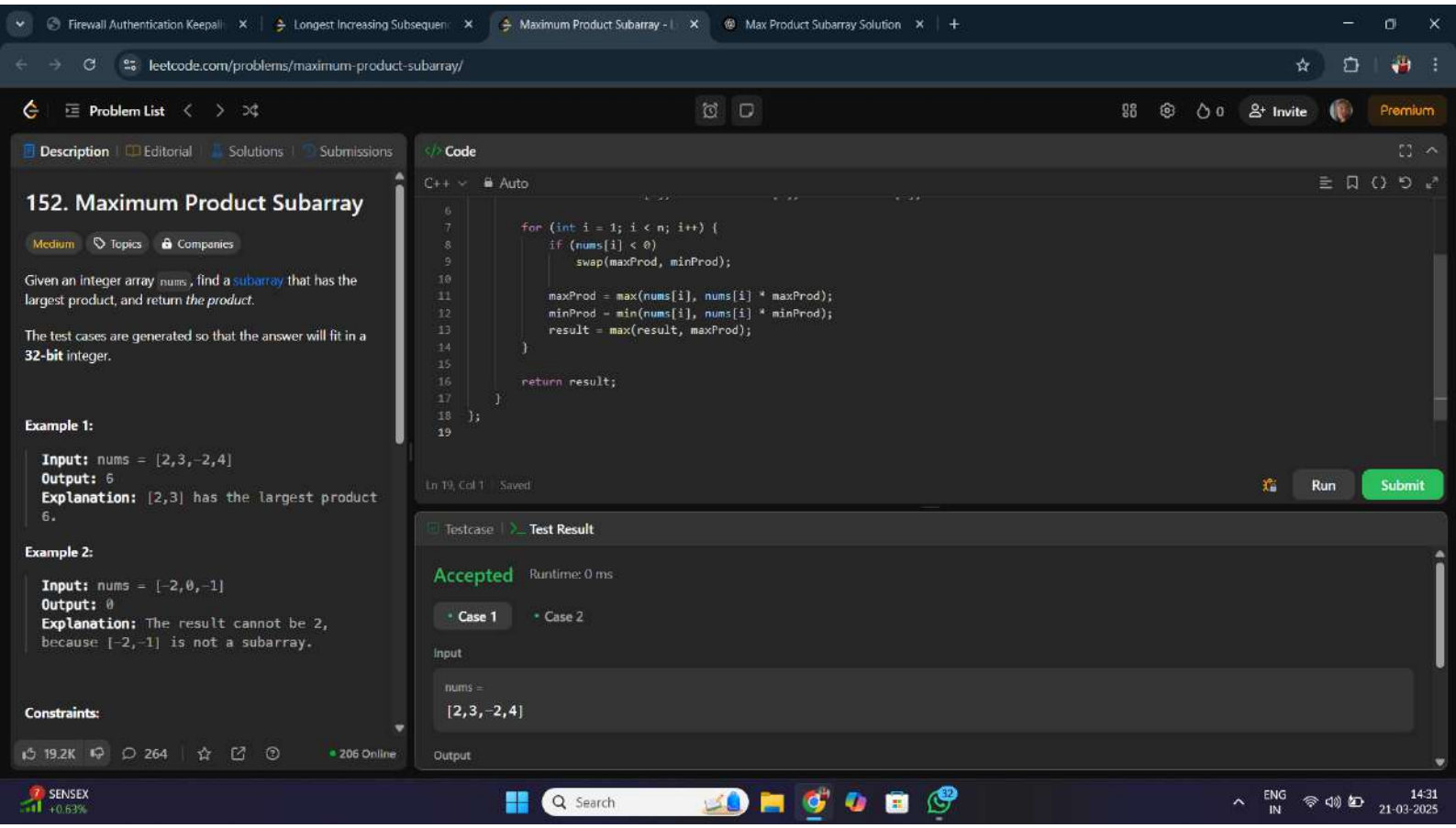
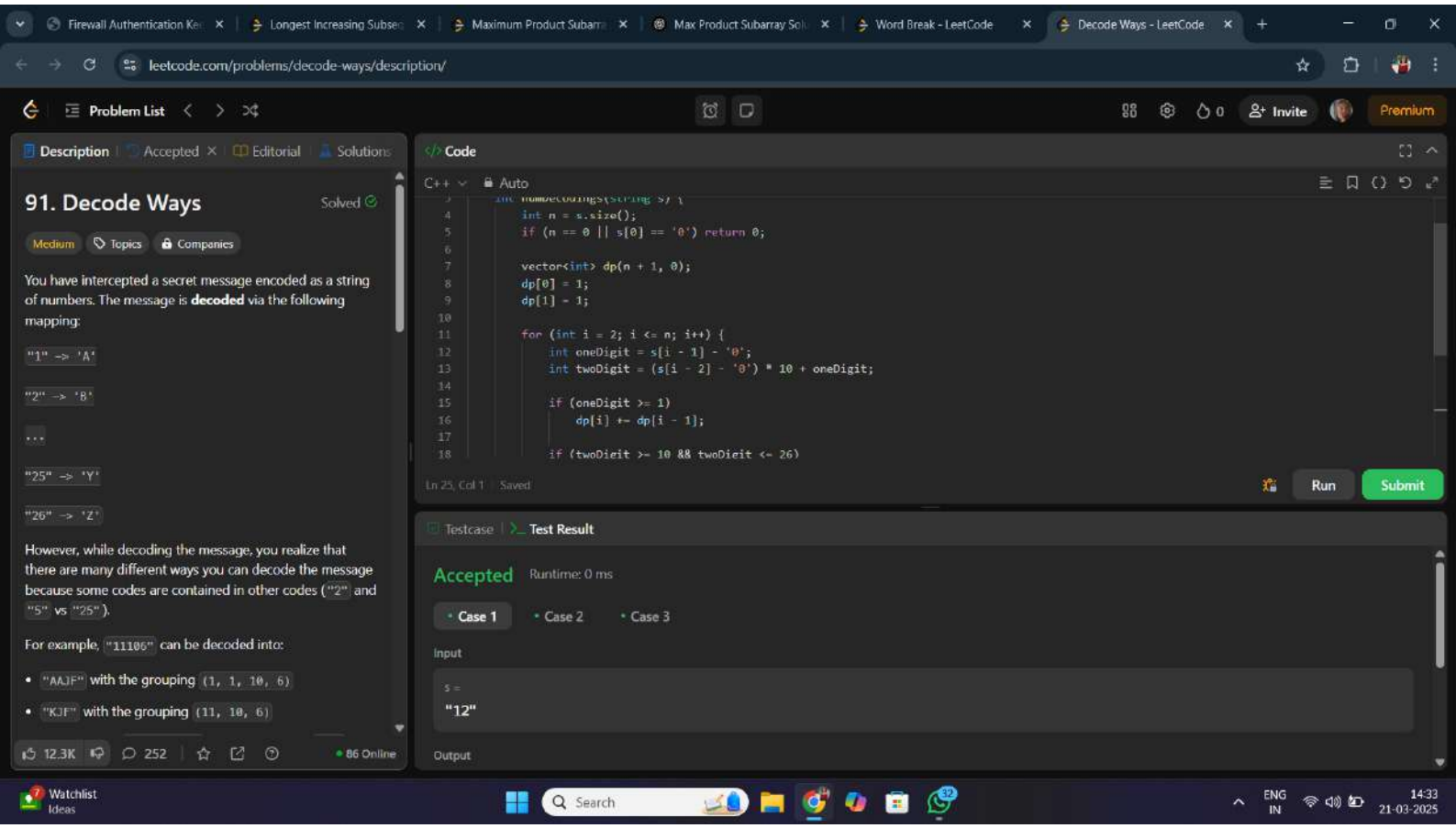
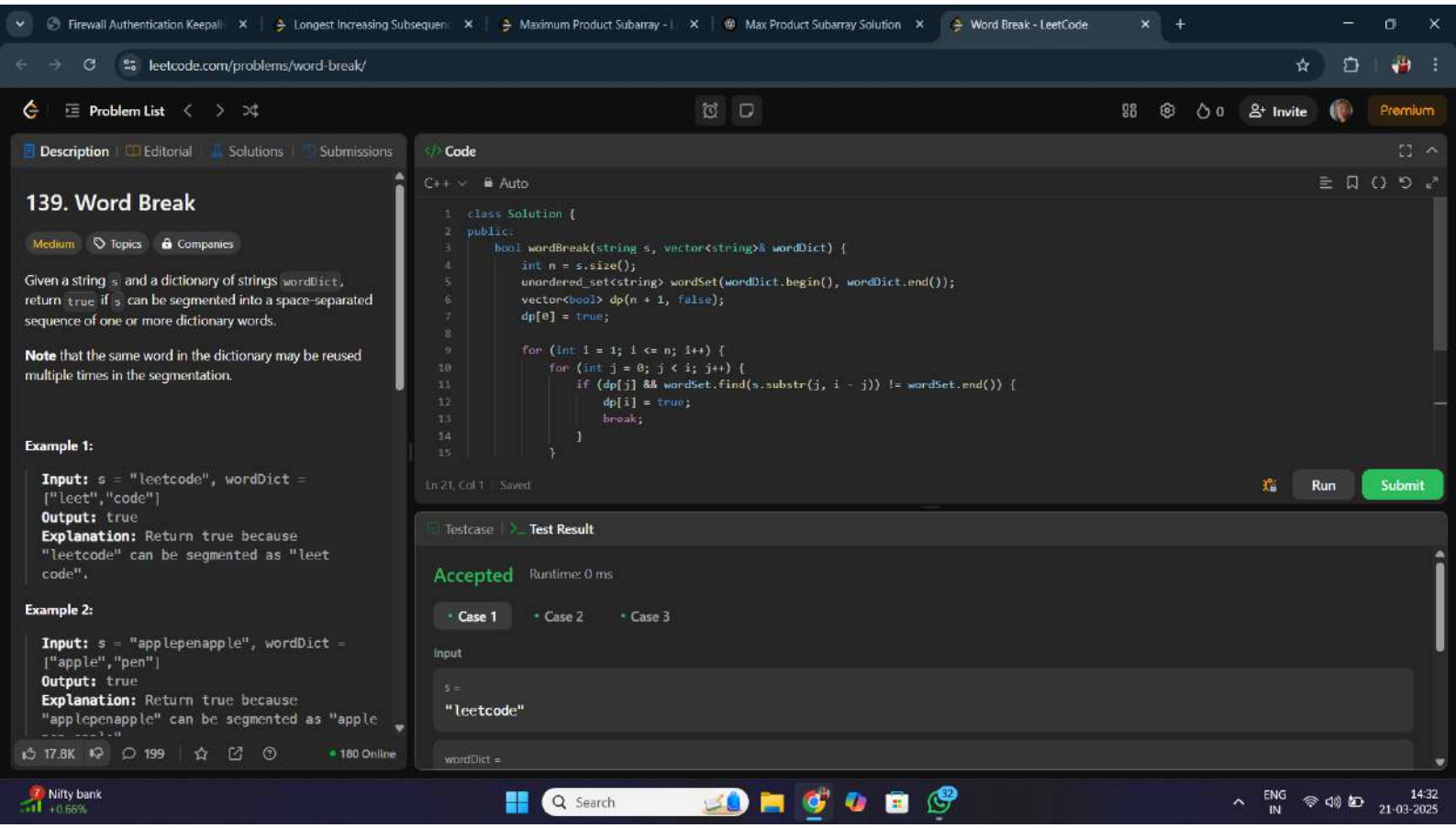

Firewall Authentication Keepali
Longest Increasing Subsequenc
leetcode.com/problems/longest-increasing-subsequence/
Problem List
Invite
Premium
Description
Editorial
Solutions
Submissions
300. Longest Increasing Subsequence
Medium
Topics
Companies
Given an integer array nums, return the length of the 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
21.6K
217
255 Online
Code
C++
Auto
class Solution {
public:
int lengthOfLIS(vector<int>& nums) {
if (nums.empty()) return 0;
vector<int> dp(nums.size(), 1); // dp[i] represents the LIS ending at index i
int maxLength = 1;
for (int i = 1; i < nums.size(); ++i) {
for (int j = 0; j < i; ++j) {
if (nums[i] > nums[j]) {
dp[i] = max(dp[i], dp[j] + 1);
}
}
maxLength = max(maxLength, dp[i]);
Ln 1, Col 1
Saved
Run
Submit
Testcase
Test Result
Run Ctrl '
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
nums =
[10,9,2,5,3,7,101,18]
Output
News for you
YouTube partne...
Search
ENG
IN
14:29
21-03-2025


Firewall Authentication Keepali
Longest Increasing Subsequenc
Maximum Product Subarray - L
Max Product Subarray Solution
leetcode.com/problems/maximum-product-subarray/
Problem List
Invite
Premium
Description
Editorial
Solutions
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:
19.2K
264
206 Online
Code
C++
Auto
for (int i = 1; i < n; 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;
}
};
Ln 19, Col 1
Saved
Run
Submit
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
nums =
[2,3,-2,4]
Output
SENSEX
+0.63%
Search
ENG
IN
14:31
21-03-2025


Firewall Authentication Ke
Longest Increasing Subseq
Maximum Product Subarra
Max Product Subarray Solu
Word Break - LeetCode
Decode Ways - LeetCode
leetcode.com/problems/decode-ways/description/
Problem List
Invite
Premium
Description
Accepted
Editorial
Solutions
91. Decode Ways
Solved
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)
12.3K
252
86 Online
Code
C++
Auto
int n = s.size();
if (n == 0 || s[0] == '0') return 0;
vector<int> dp(n + 1, 0);
dp[0] = 1;
dp[1] = 1;
for (int i = 2; i <= n; i++) {
int oneDigit = s[i - 1] - '0';
int twoDigit = (s[i - 2] - '0') * 10 + oneDigit;
if (oneDigit >= 1)
dp[i] += dp[i - 1];
if (twoDigit >= 10 && twoDigit <= 26)
Ln 25, Col 1
Saved
Run
Submit
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
s =
"12"
Output
Watchlist
Ideas
Search
ENG
IN
14:33
21-03-2025


Firewall Authentication Keepali
Longest Increasing Subsequenc
Maximum Product Subarray - I
Max Product Subarray Solution
Word Break - LeetCode
leetcode.com/problems/word-break/
Problem List
Invite
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
17.8K
199
180 Online
Code
C++
Auto
class Solution {
public:
bool wordBreak(string s, vector<string>& wordDict) {
int n = s.size();
unordered_set<string> wordSet(wordDict.begin(), wordDict.end());
vector<bool> dp(n + 1, false);
dp[0] = true;
for (int i = 1; i <= n; i++) {
for (int j = 0; j < i; j++) {
if (dp[j] && wordSet.find(s.substr(j, i - j)) != wordSet.end()) {
dp[i] = true;
break;
}
}
Ln 21, Col 1
Saved
Run
Submit
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
s =
"leetcode"
wordDict =
Nifty bank
+0.66%
Search
ENG
IN
14:32
21-03-2025