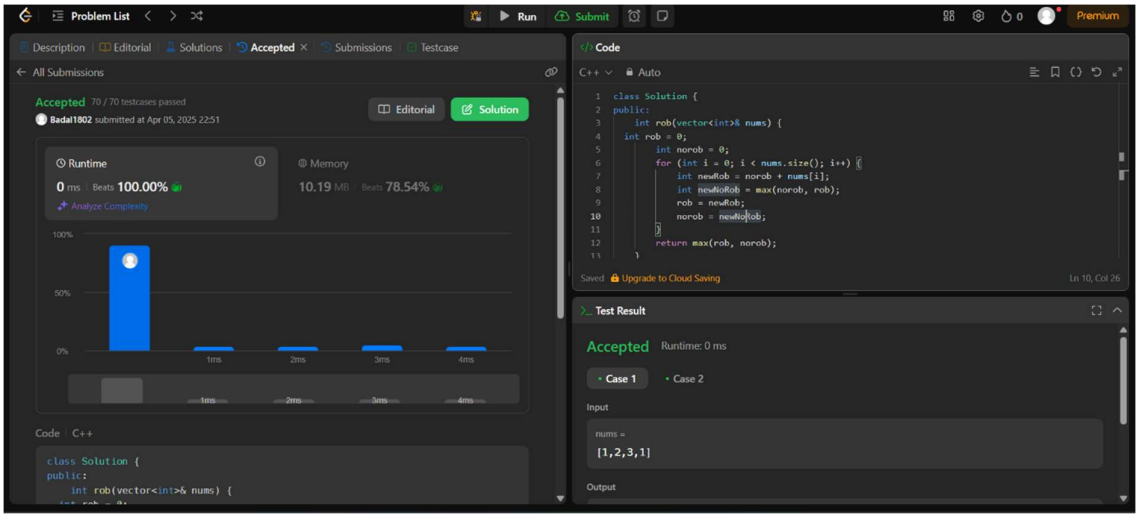
Experiment-7

House Robber

CODE:

```
class Solution {
public:
    int rob(vector<int>& nums) {
  int rob = 0;
        int norob = 0;
        for (int i = 0; i < nums.size(); i++) {
            int newRob = norob + nums[i];
            int newNoRob = max(norob, rob);
            rob = newRob;
            norob = newNoRob;
        }
        return max(rob, norob);
    }
};
```

OUTPUT:

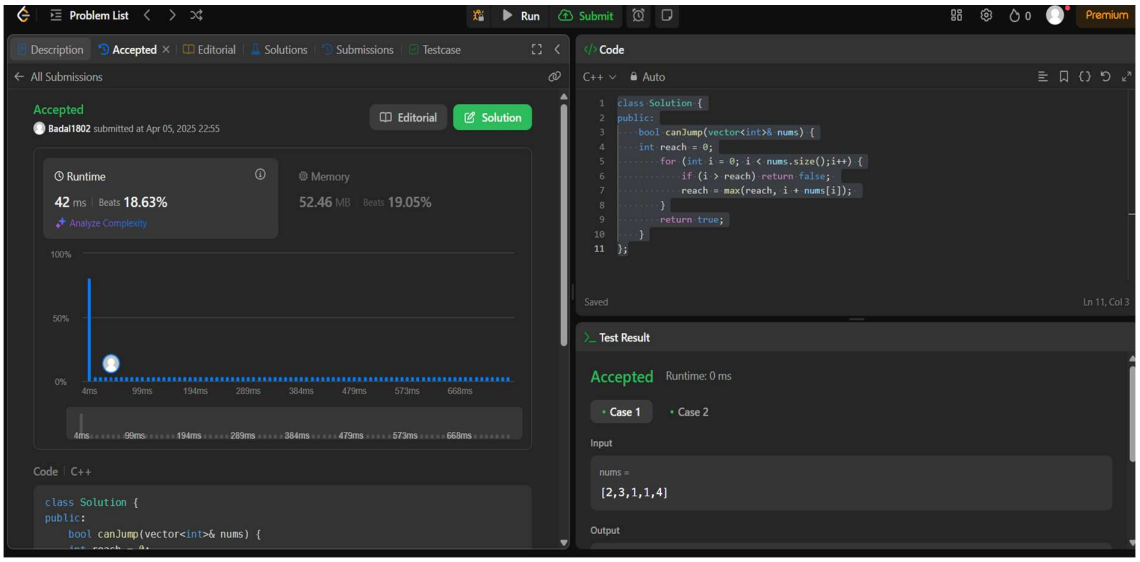


Jump Game

CODE:

```
class Solution {
public:
    bool canJump(vector<int>& nums) {
    int reach = 0;
        for (int i = 0; i < nums.size();i++) {
            if (i > reach) return false;
            reach = max(reach, i + nums[i]);
        }
        return true;
    }
};
```

OUTPUT:

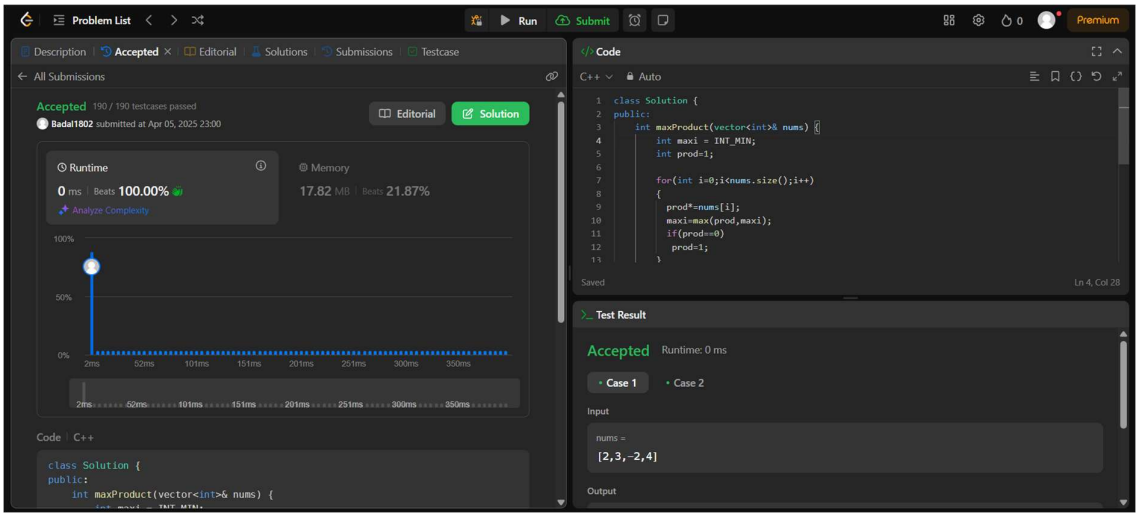


Maximum Product Subarray

CODE:

```
class Solution {
public:
    int maxProduct(vector<int>& nums) {
        int maxi = INT_MIN;
        int prod=1;
        for(int i=0;i<nums.size();i++)
        {
          prod*=nums[i];
          maxi=max(prod,maxi);
          if(prod==0)
           prod=1;
        }
        prod=1;
        for(int i=nums.size()-1;i>=0;i--)
        {
          prod*=nums[i];
          maxi=max(prod,maxi);
          if(prod==0)
           prod=1;
        }
        return maxi;
    }
};
```

OUTPUT:



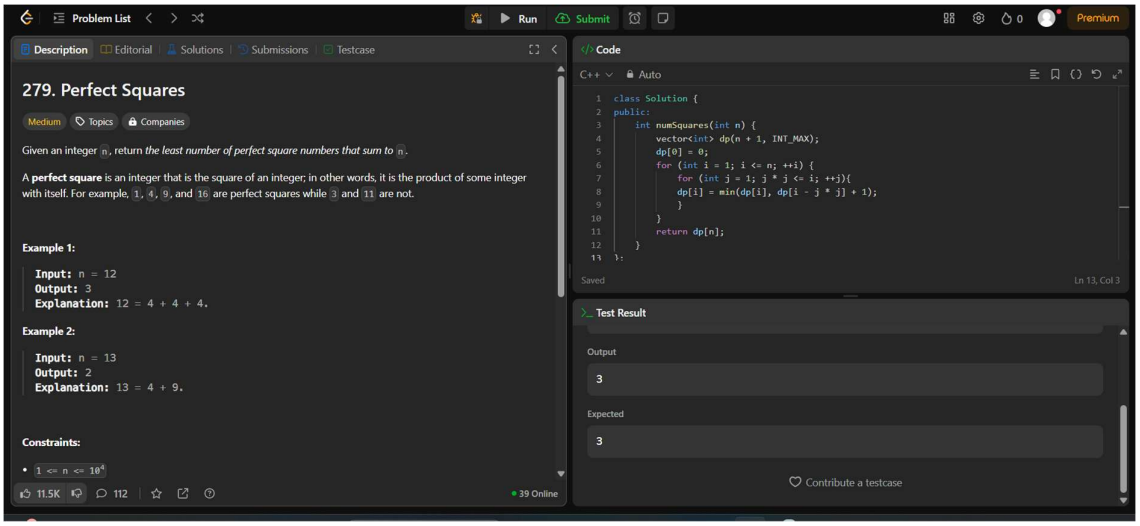
Perfect Squares

CODE:

```
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];
    }
```

};

OUTPUT:


Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
Testcase
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
39 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 13, Col 3
Test Result
Output
3
Expected
3
Contribute a testcase