

AP Assignment 7

**aliza asif**

22BCS50175

22BCS\_FL\_IOT\_601\_A

**AP ASSIGNMENT 7**

**Q1.** **HOUSE ROBBER**

Implementation Code:

class Solution {

public:

int rob(vector<int>& nums) {

int n = nums.size();

if (n == 0) return 0;

if (n == 1) return nums[0];

vector<int> dp(n);

dp[0] = nums[0];

dp[1] = max(nums[0], nums[1]);

for (int i = 2; i < n; i++) {

dp[i] = max(dp[i - 1], dp[i - 2] + nums[i]);

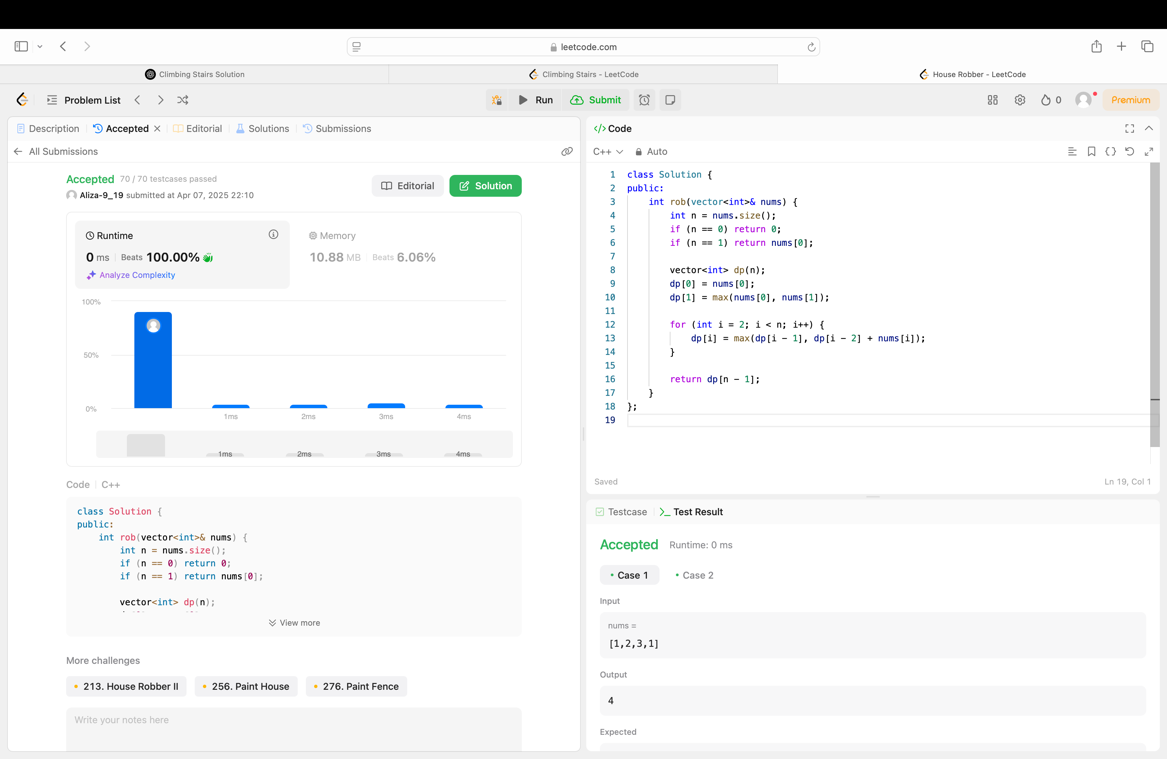
}

return dp[n - 1];

}

};

Output:



**Q2. JUMP GAMES**

Implementation Code:

class Solution {

public:

bool canJump(vector<int>& nums) {

int maxReach = 0;

for (int i = 0; i < nums.size(); i++) {

if (i > maxReach) return false;

maxReach = max(maxReach, i + nums[i]);

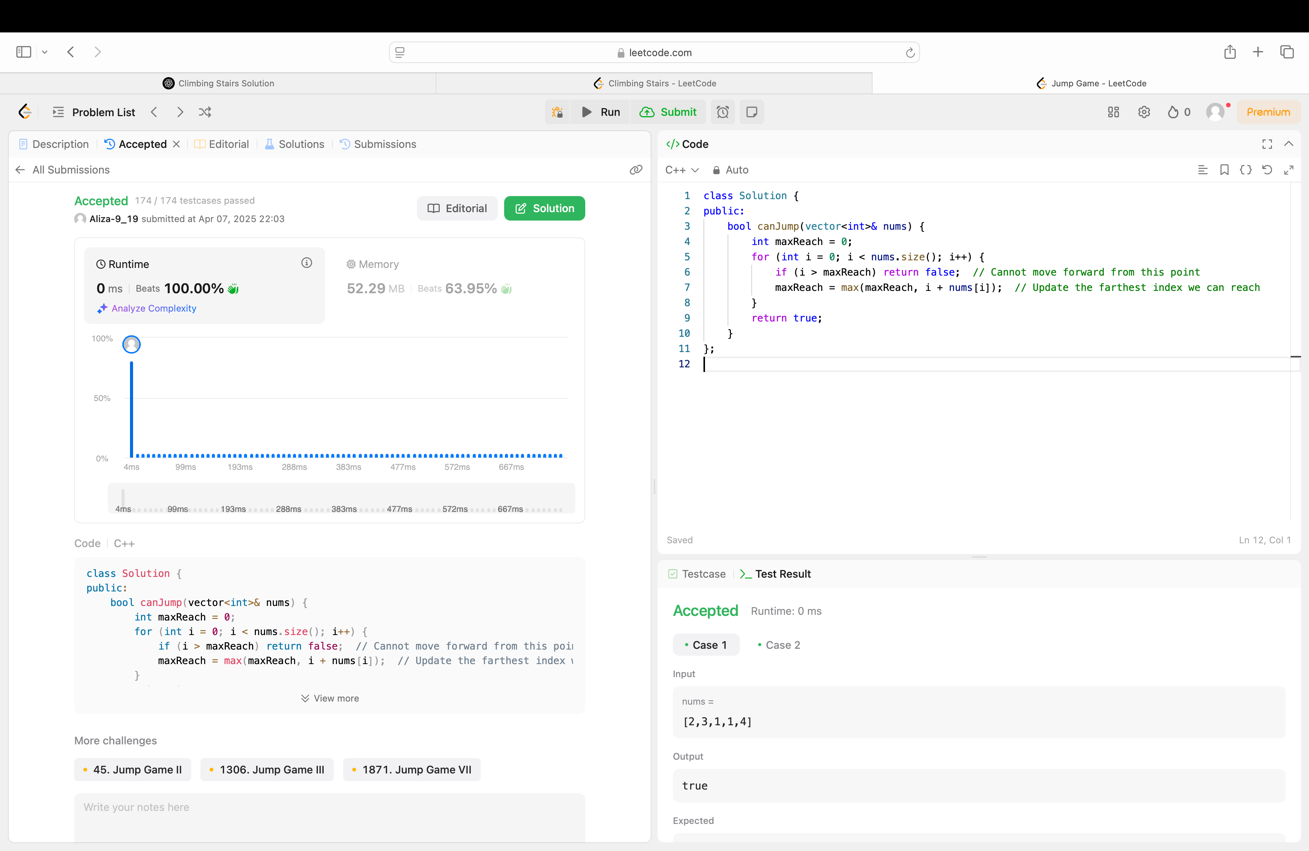
}

return true;

}

};

Output:



**Q3. Maximum product subarray**

Implementation Code:

class Solution {

public:

int maxProduct(vector<int>& nums) {

int maxProd = nums[0];

int currMax = nums[0];

int currMin = nums[0];

for (int i = 1; i < nums.size(); i++) {

if (nums[i] < 0) {

swap(currMax, currMin);

}

currMax = max(nums[i], currMax \* nums[i]);

currMin = min(nums[i], currMin \* nums[i]);

maxProd = max(maxProd, currMax);

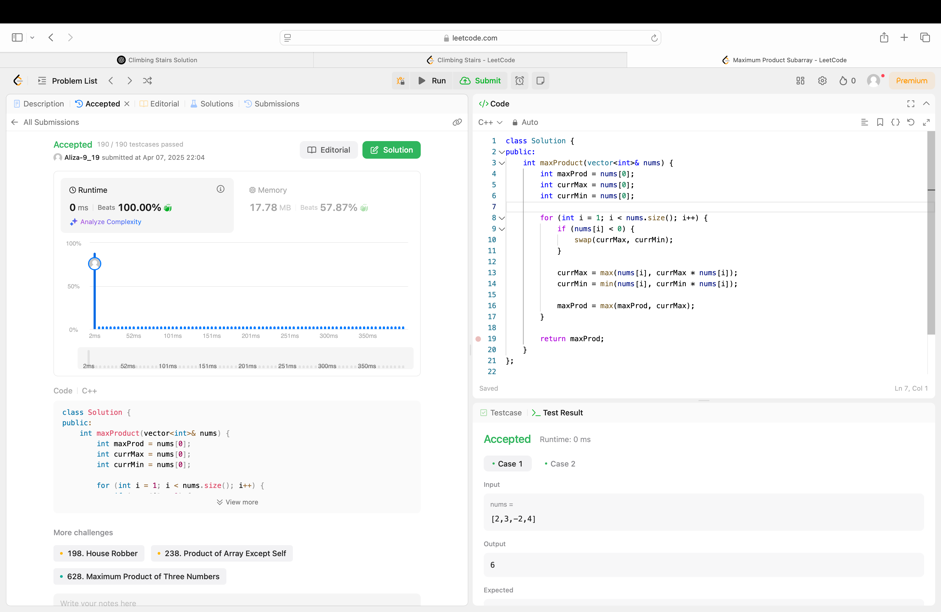
}

return maxProd;

}

};

Output:



**Q4. Maximum Subarray (53)**

Implementation 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:

