**ASSIGNMENT 4**

1. [**1763. Longest Nice Substring**](https://leetcode.com/problems/longest-nice-substring/)

class Solution {

public String longestNiceSubstring(String s) {

Set<Character> charSet = new HashSet<>();

for (int i = 0; i < s.length(); i++) {

charSet.add(s.charAt(i));

}

for (int i = 0; i < s.length(); i++) {

if (charSet.contains(Character.toUpperCase(s.charAt(i))) &&

charSet.contains(Character.toLowerCase(s.charAt(i)))) {

continue;

}

String s1 = longestNiceSubstring(s.substring(0, i));

String s2 = longestNiceSubstring(s.substring(i+1));

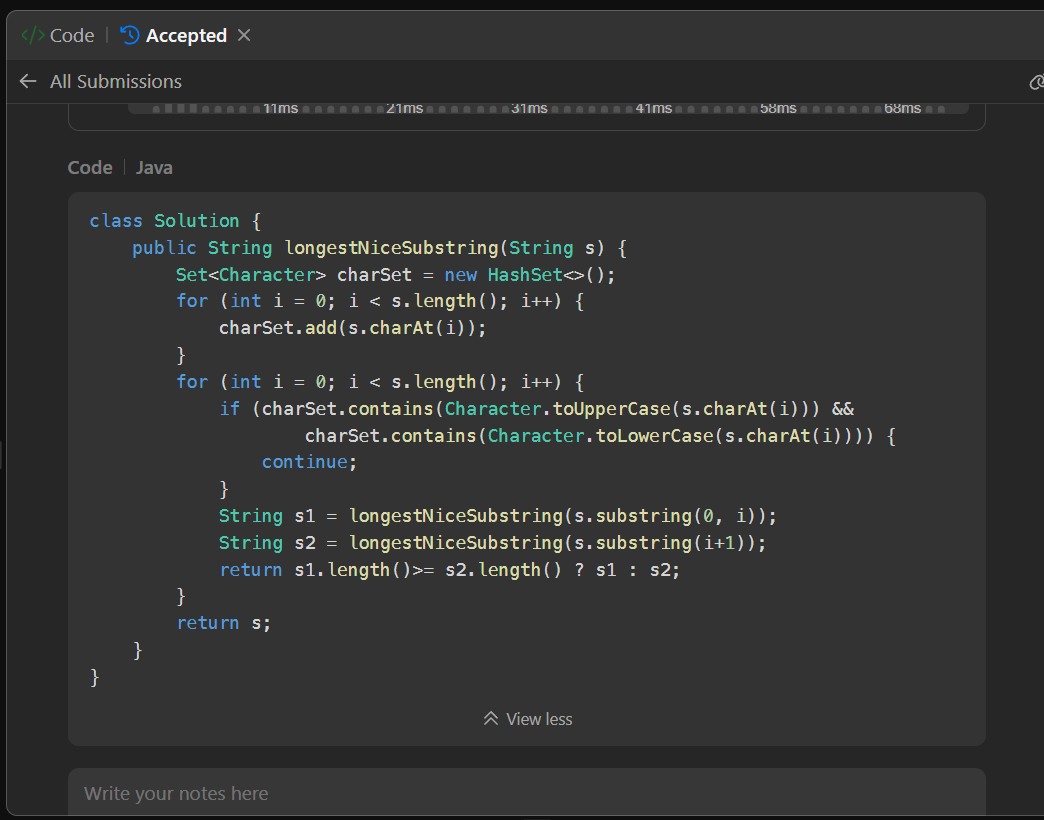
return s1.length()>= s2.length() ? s1 : s2;

}

return s;

}

}



1. **190.**[**Reverse Bits**](https://leetcode.com/problems/reverse-bits/description/)

public class Solution {

// you need treat n as an unsigned value

public int reverseBits(int n) {

if (n == 0) return 0;

int result = 0;

for (int i = 0; i < 32; i++) {

result <<= 1;

result += n & 1;

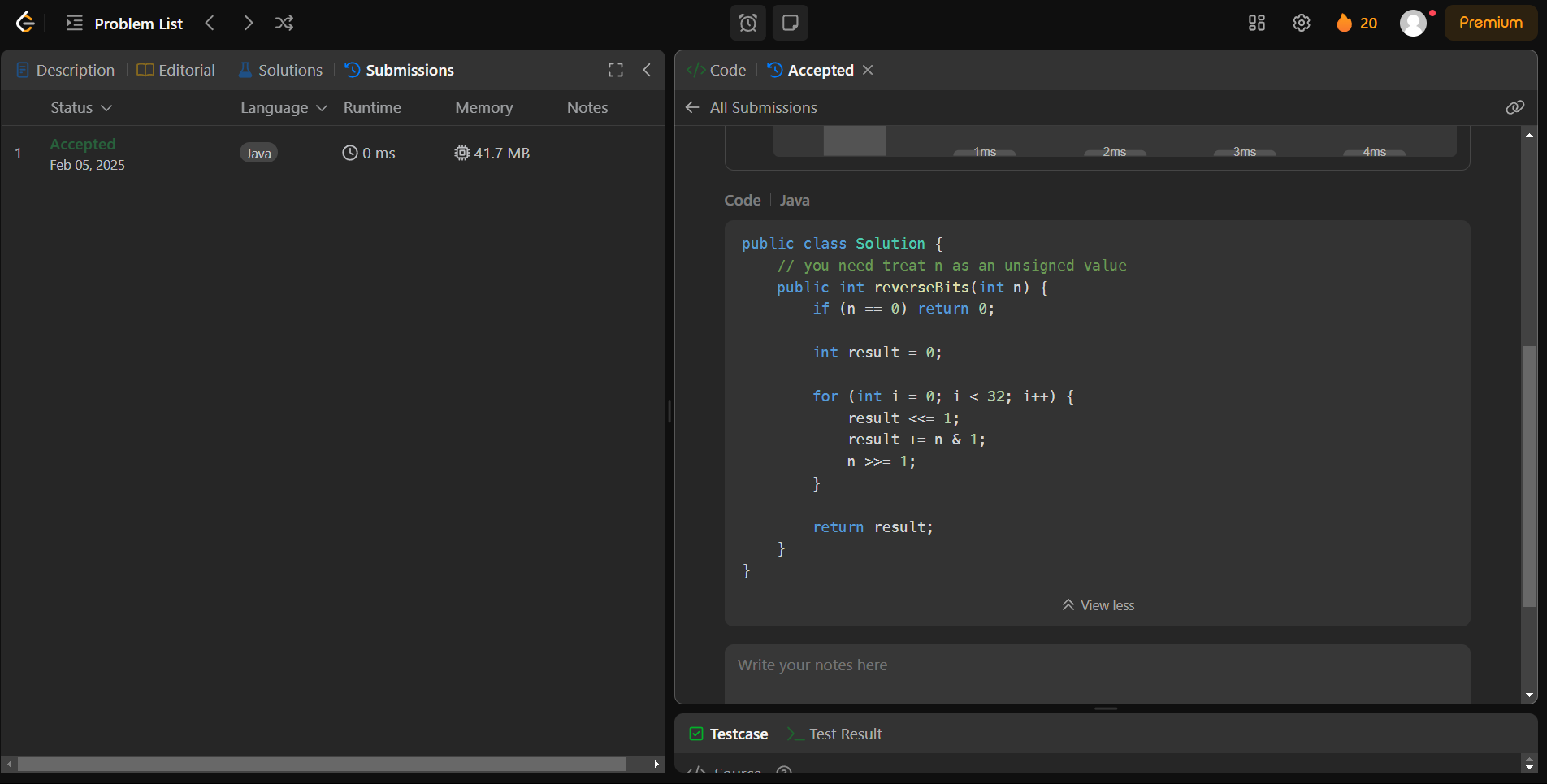
n >>= 1;

}

return result;

}

}



1. **191.**[**Number of 1 Bits**](https://leetcode.com/problems/number-of-1-bits/description/)

public class Solution {

public int hammingWeight(int n) {

int res = 0;

for (int i = 0; i < 32; i++) {

if (((n >> i) & 1) == 1) {

res += 1;

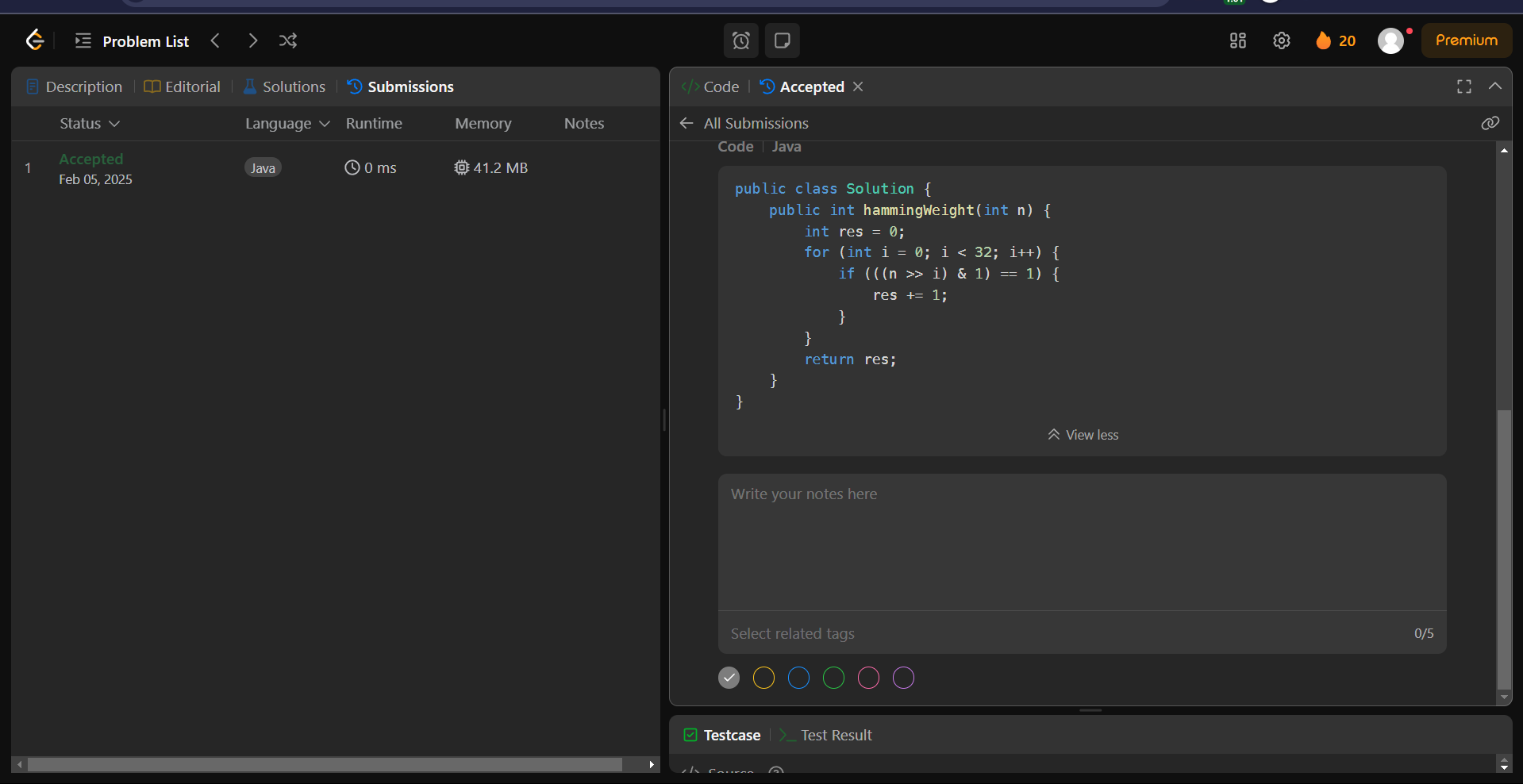
}

}

return res;

}

}

****

1. **53.**[**Maximum Subarray**](https://leetcode.com/problems/maximum-subarray/description/)

class Solution {

public int maxSubArray(int[] nums) {

int maxi = Integer.MIN\_VALUE;

int sum = 0;

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

sum = Math.max(nums[i], sum + nums[i]);

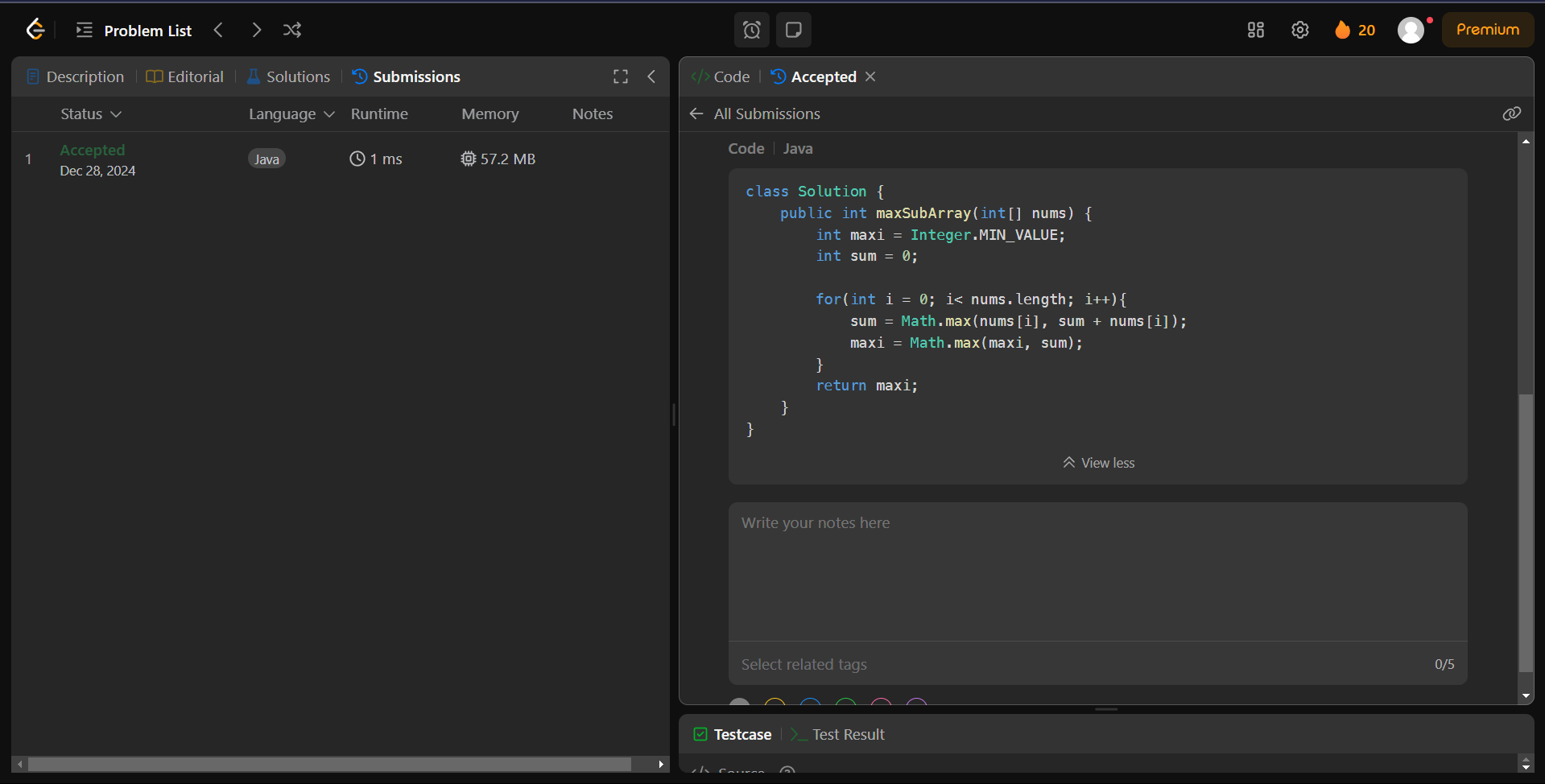
maxi = Math.max(maxi, sum);

}

return maxi;

}

}

****

1. **240.**[**Search a 2D Matrix II**](https://leetcode.com/problems/search-a-2d-matrix-ii/description/)

class Solution {

public boolean searchMatrix(int[][] matrix, int target) {

int n = matrix.length, m = matrix[0].length;

int row = 0, col = m-1;

while(row < n && col >= 0){

if(matrix[row][col] == target) return true;

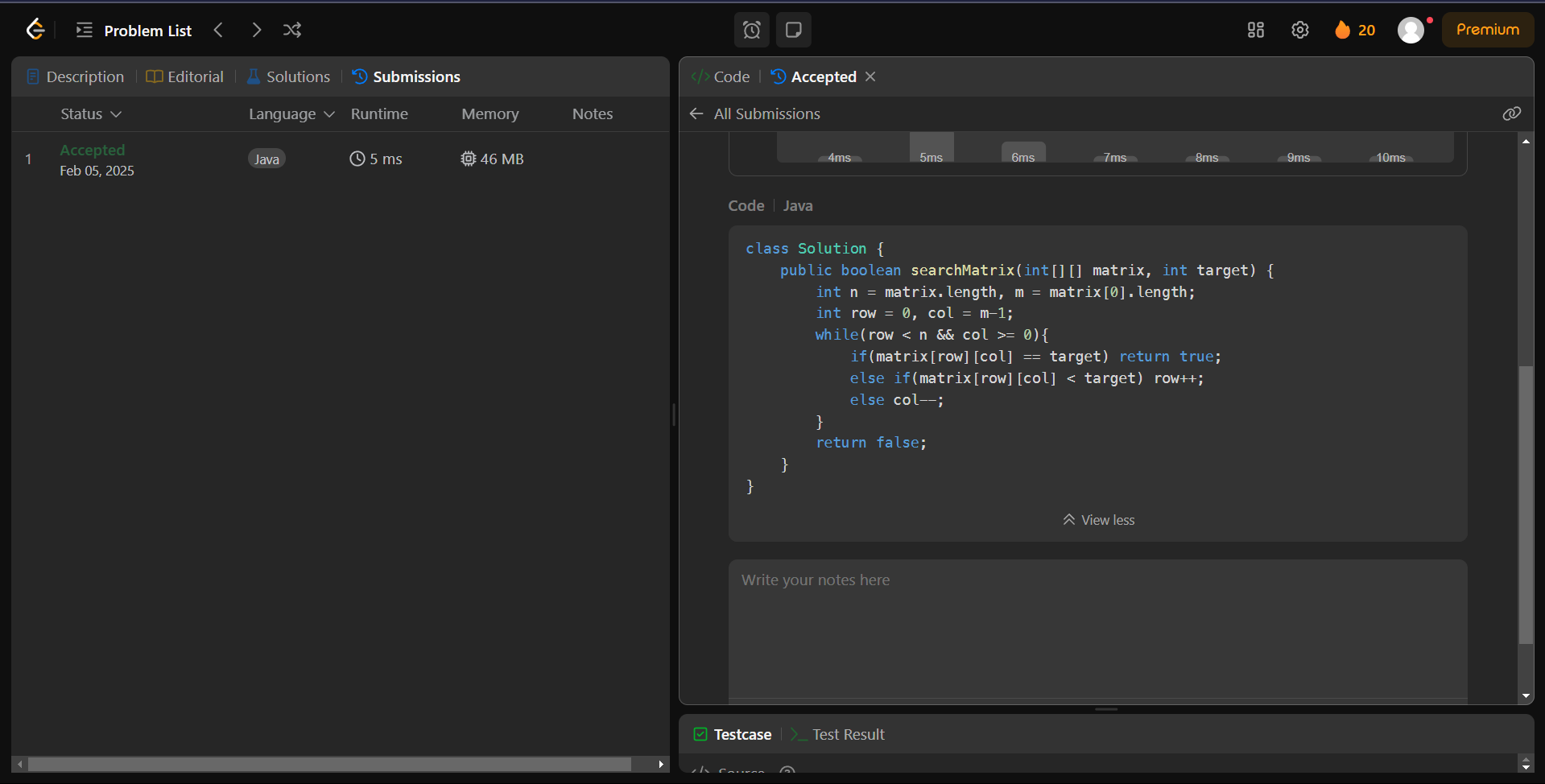
else if(matrix[row][col] < target) row++;

else col--;

}

return false;

}

}

1. **372.**[**Super Pow**](https://leetcode.com/problems/super-pow/description/)

class Solution {

private int binExp(int a, int b, int M) {

int res = 1;

a %= M;

while (b > 0) {

if ((b & 1) != 0)

res = (res \* a) % M;

a = (a \* a) % M;

b >>= 1;

}

return res;

}

public int superPow(int a, int[] b) {

int m = 1140;

int exp = 0;

for (int i = 0; i < b.length; i++)

exp = (exp \* 10 + b[i]) % m;

if (exp == 0)

exp = 1140

return binExp(a, exp, 1337);

}

}

