**Assignment 4**

**Name – Aditya Taragi Uid – 22BCS10422.**

**Problem 1: Longest Nice Substring (**<https://leetcode.com/problems/longest-nice-substring/> **)**

**Code:**class Solution {

public:

    string longestNiceSubstring(string s) {

        if(s.size() < 2) return "";

        unordered\_set<char> charSet(s.begin(),s.end());

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

            if(charSet.count(tolower(s[i])) == 0 || charSet.count(toupper(s[i])) == 0) {

                string left = longestNiceSubstring(s.substr(0,i));

                string right = longestNiceSubstring(s.substr(i+1));

                return (left.size() >= right.size()) ? left : right;

            }

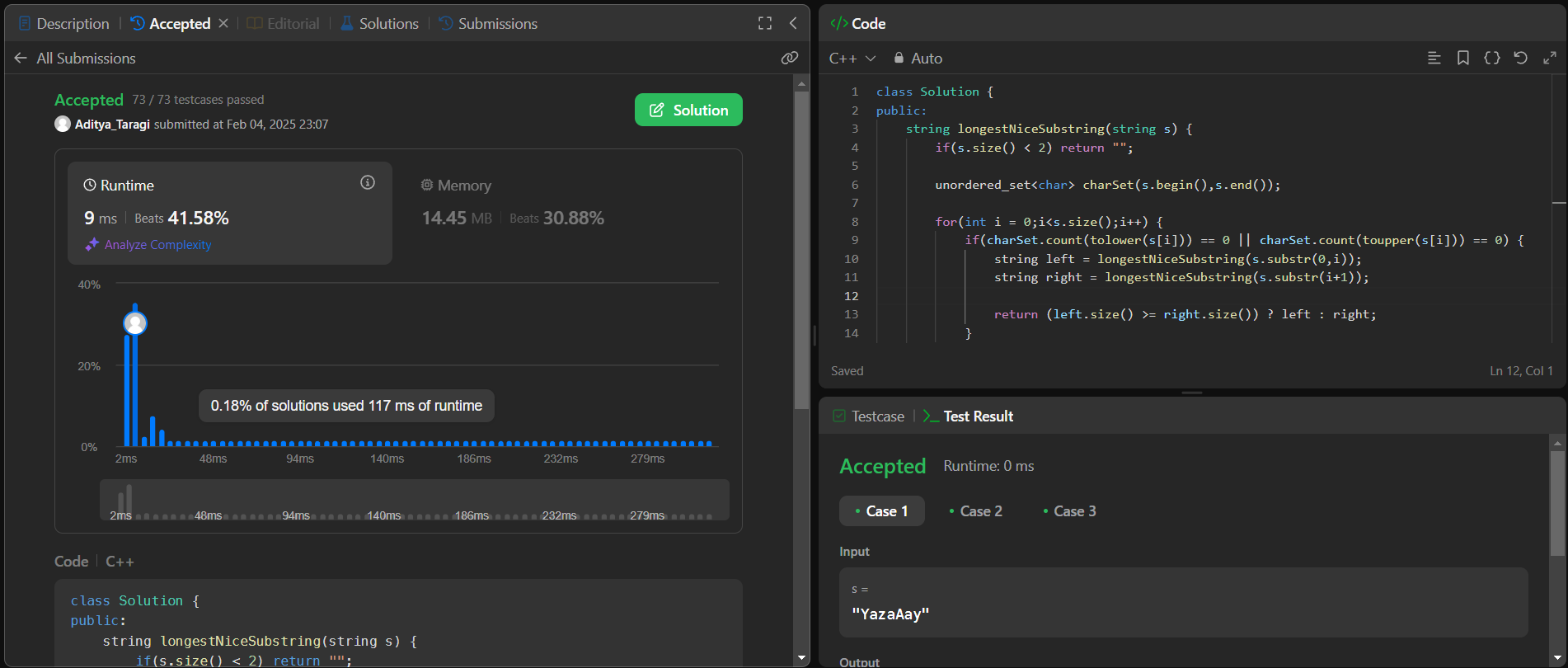
        }

        return s;

    }

};

**Screenshot:**

****

**Problem 2: Reverse Bits (**<https://leetcode.com/problems/reverse-bits/> )

**Code:**class Solution {

public:

    uint32\_t reverseBits(uint32\_t n) {

        uint32\_t result = 0;

        for(int i = 0;i<32;i++) {

            uint32\_t temp = n & 1;

            result = (result << 1) | temp;

            n = n >> 1;

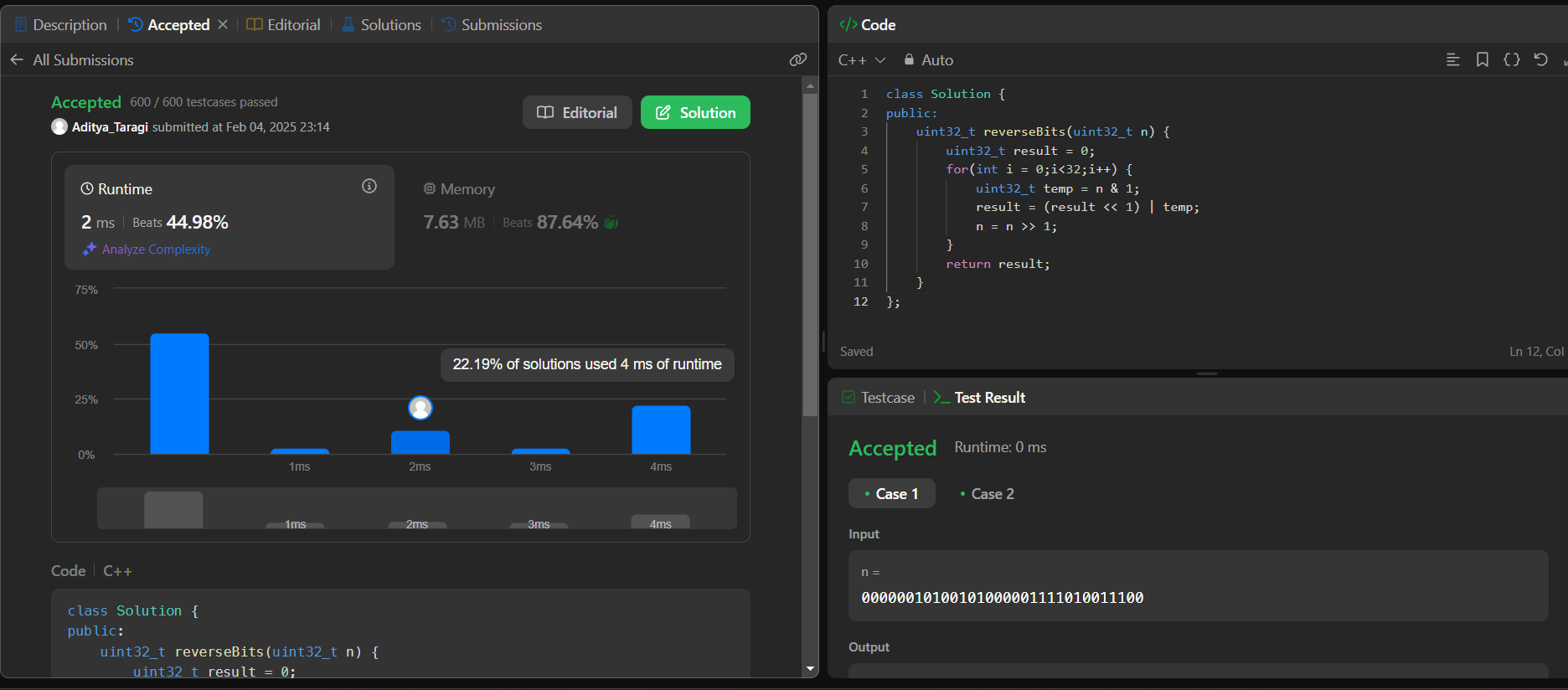
        }

        return result;

    }

};

**Screenshot:**



**Problem 3: Number of 1 Bits (**<https://leetcode.com/problems/number-of-1-bits/> **)**

**Code:**class Solution {

public:

    int hammingWeight(int n) {

        int count = 0;

        while(n) {

            n = n & (n-1);

            count++;

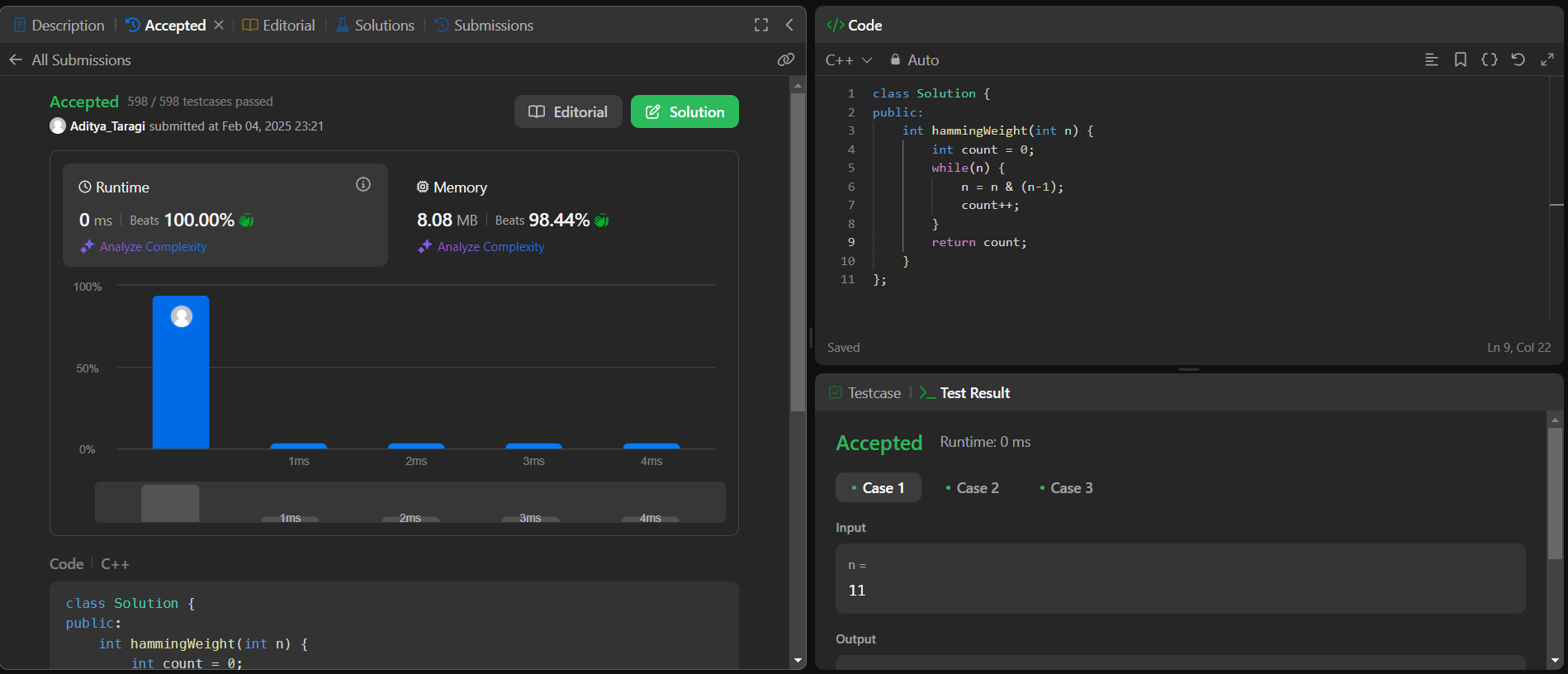
        }

        return count;

    }

};

**Screenshot:**



**Problem 4: Maximum Subarray (**<https://leetcode.com/problems/maximum-subarray/> **)**

**Code:**class Solution {

public:

    int maxSubArray(vector<int>& nums) {

        int sum = nums[0];

        int max\_sum = nums[0];

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

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

            max\_sum = max(sum,max\_sum);

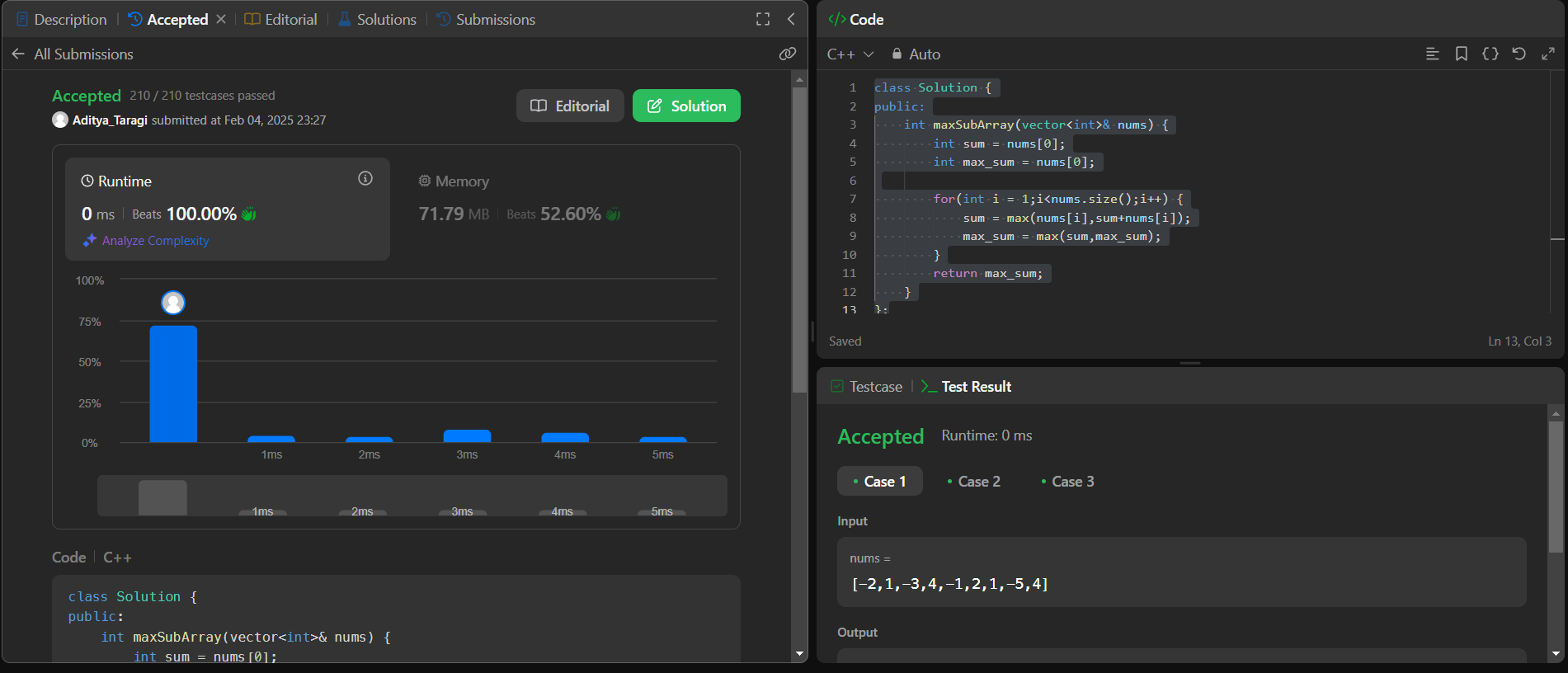
        }

        return max\_sum;

    }

};

**Screenshot:**



**Problem 5: Search a 2D matrix (**<https://leetcode.com/problems/search-a-2d-matrix-ii/> **)**

**Code:**class Solution {

public:

    bool searchMatrix(vector<vector<int>>& matrix, int target) {

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

            for(int j = 0;j<matrix[0].size();j++) {

                if(matrix[i][j] == target) {

                    return true;

                }

            }

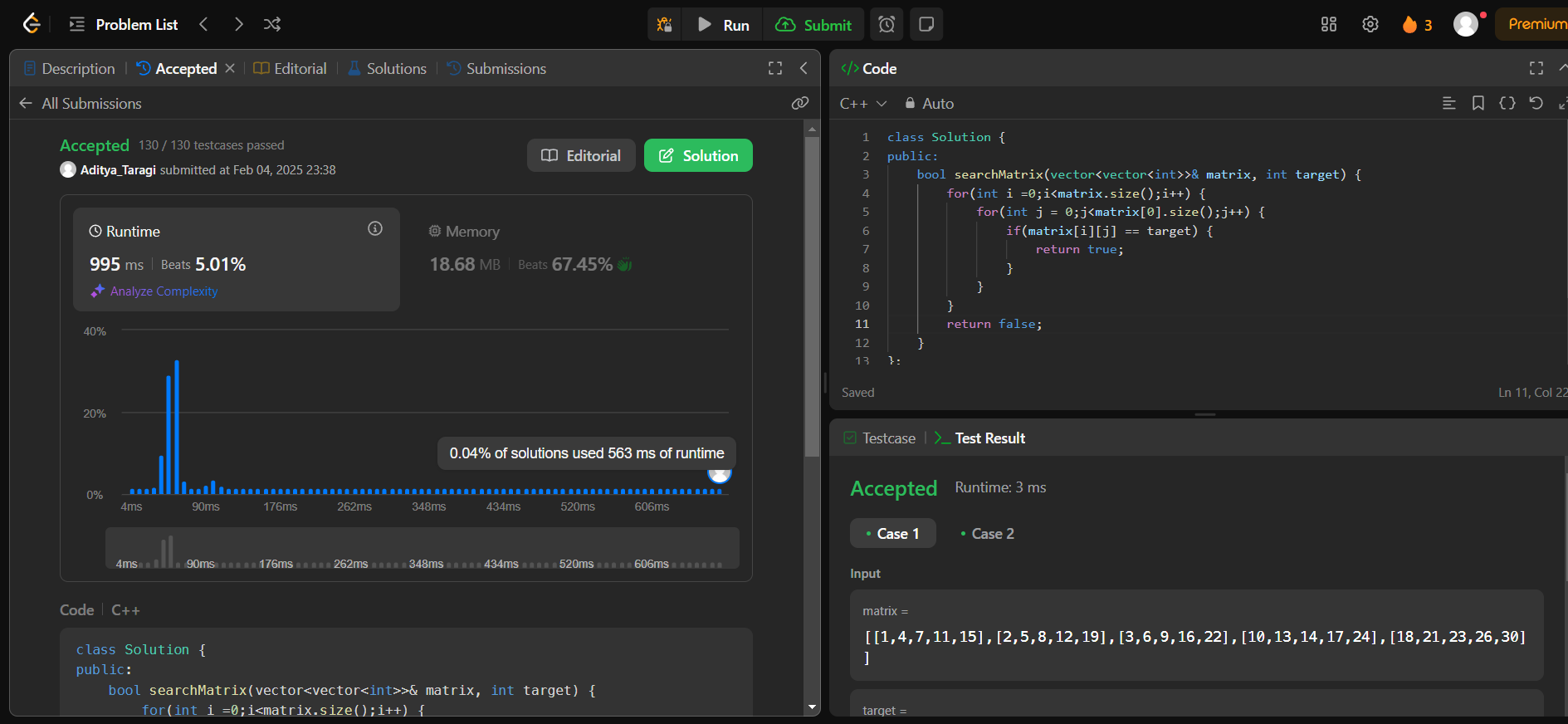
        }

        return false;

    }

};

**Screenshot:**

**Problem 6: Super Pow (**<https://leetcode.com/problems/super-pow/> **)**

**Code:**class Solution {

public:

    int modExp(int a, int b, int mod) {

        long long result = 1;

        a = a % mod;

        while(b>0) {

            if(b%2 == 1) {

                result = (result\*a) % mod;

            }

            a = (a\*a) % mod;

            b /=2;

        }

        return (int) result;

    }

    int superPow(int a, vector<int>& b) {

        int mod = 1337;

        a = a % mod;

        int result = 1;

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

            result = modExp(result,10,mod);

            result = (result \* modExp(a,b[i],mod)) % mod;

        }

        return result;

    }

};

**Screenshot:**

