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

class Solution {

public:

    string longestNiceSubstring(string s) {

        if (s.size() < 2) return "";

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

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

             if (st.find((char) toupper(s[i])) == end(st) || st.find((char) tolower(s[i])) == end(st)) {

                string s1 = longestNiceSubstring(s.substr(0, i));

                string s2 = longestNiceSubstring(s.substr(i + 1));

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

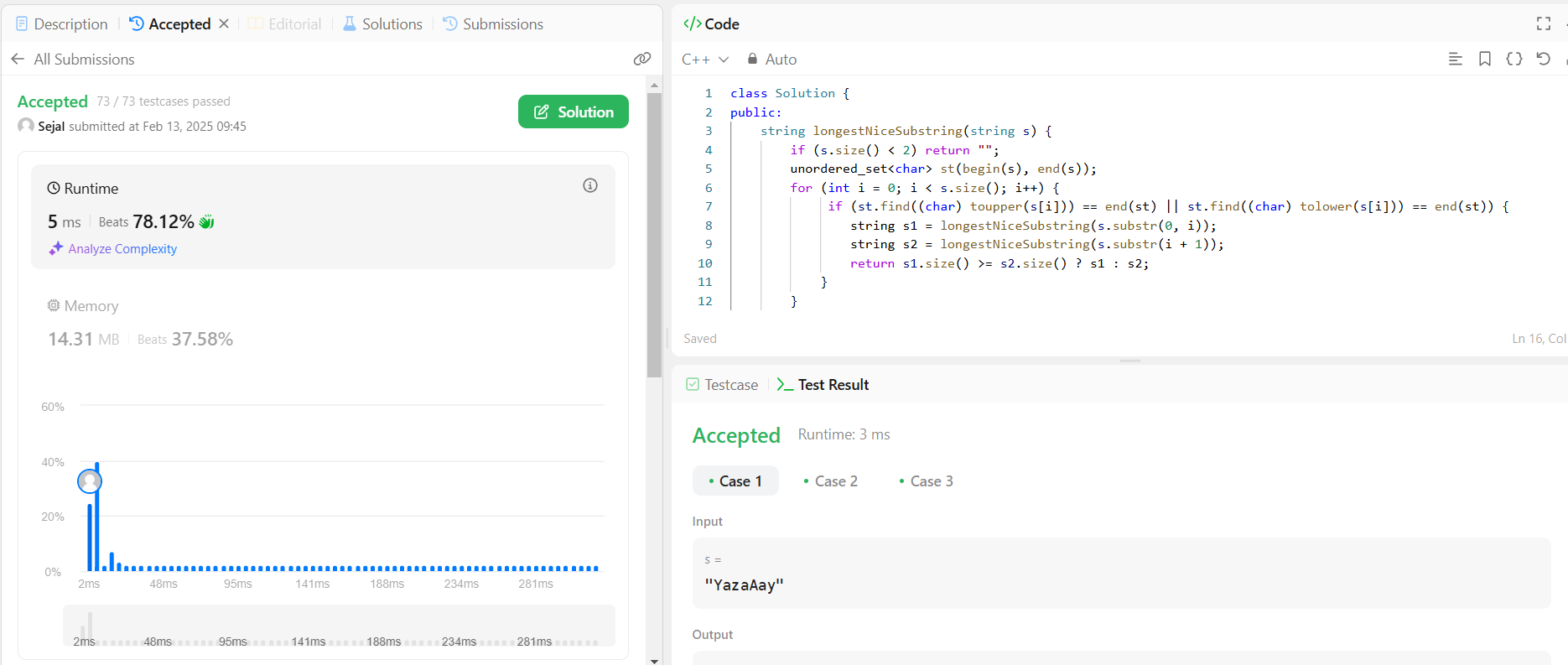
            }

        }

        return s;

    }

};



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

class Solution {

public:

    uint32\_t reverseBits(uint32\_t n) {

        string bits = bitset<32>(n).to\_string();

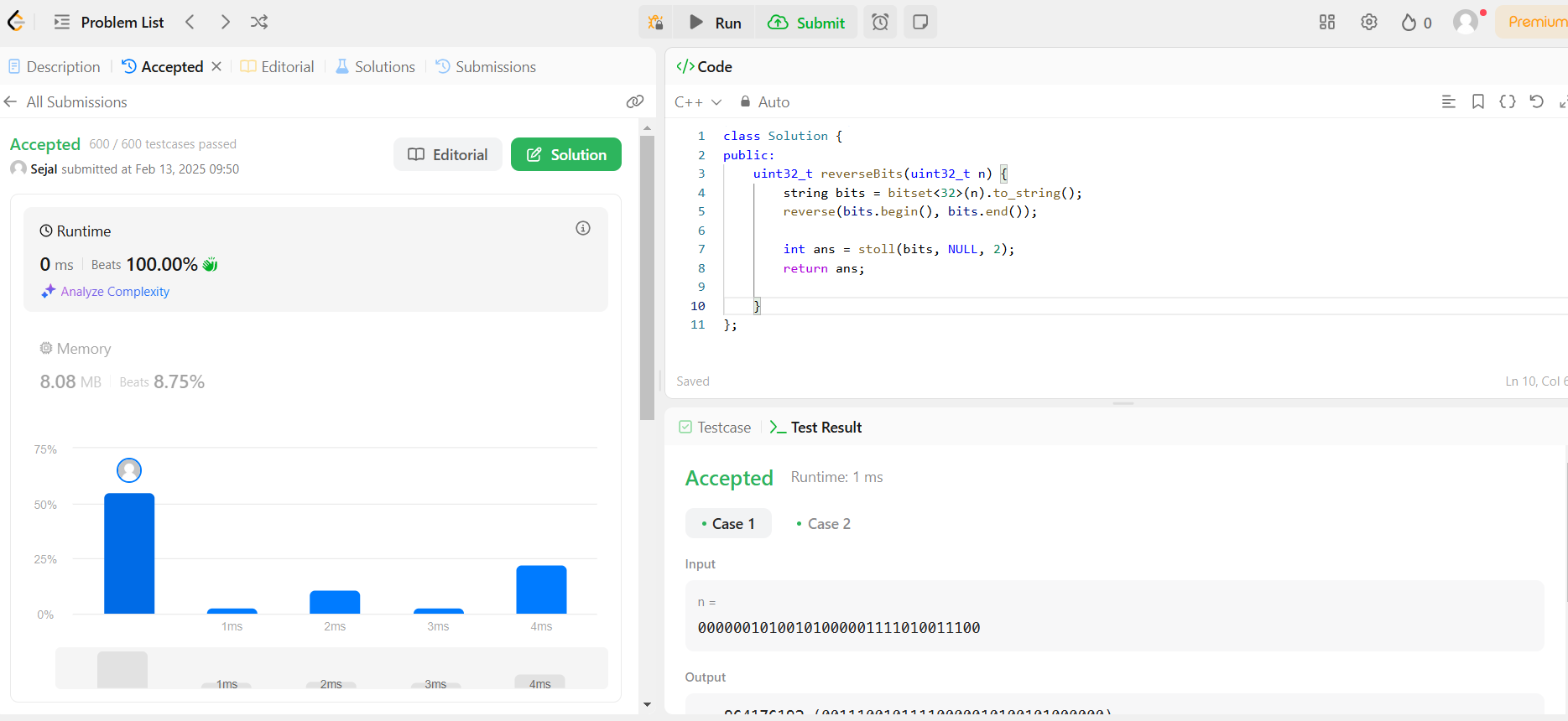
        reverse(bits.begin(), bits.end());

        int ans = stoll(bits, NULL, 2);

        return ans;

    }

};



88.[Merge Sorted Array](https://leetcode.com/problems/merge-sorted-array/description/)

class Solution {

public:

    void merge(vector<int>& nums1, int m, vector<int>& nums2, int n) {

        int i=m-1;

        int j=n-1;

        int k=m+n-1;

        while(j>=0 && i>=0){

            if(nums1[i]>nums2[j]){

                nums1[k]=nums1[i];

                i--;

                k--;

            }

            else{

                nums1[k]=nums2[j];

                k--;

                j--;

            }

        }

        while(j>=0){

            nums1[k]=nums2[j];

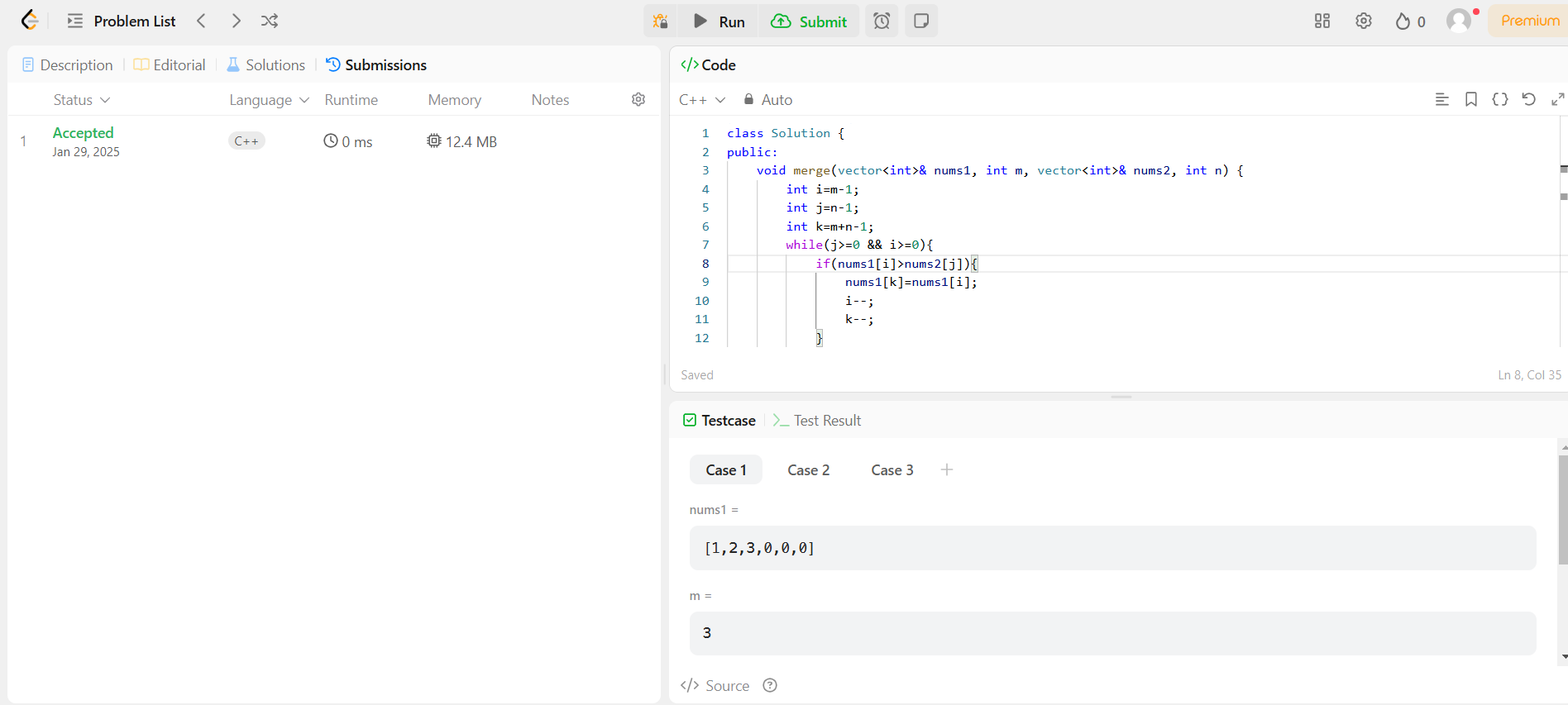
            j--;

            k--;

        }

    }

};



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

class Solution {

    const int base = 1337;

    int powmod(int a, int k)

    {

        a %= base;

        int result = 1;

        for (int i = 0; i < k; ++i)

            result = (result \* a) % base;

        return result;

    }

public:

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

        if (b.empty()) return 1;

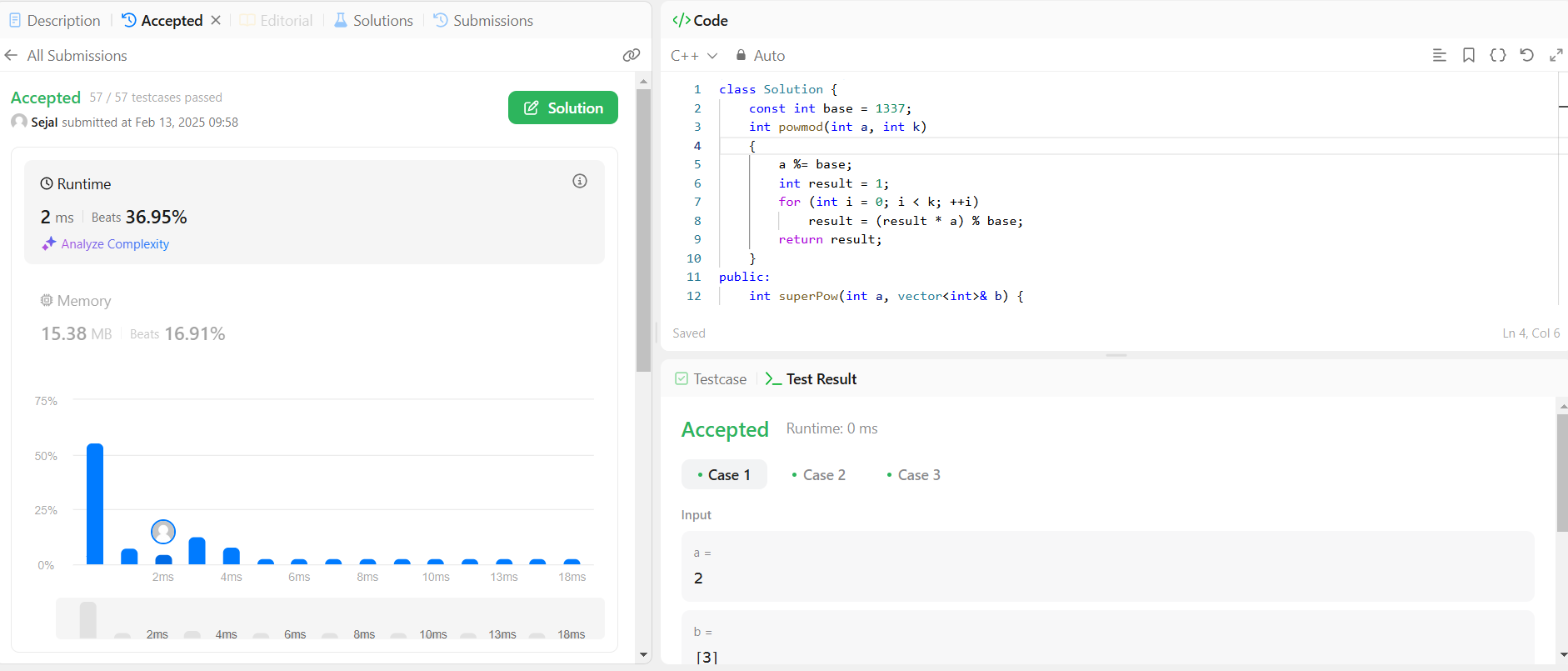
        int last\_digit = b.back();

        b.pop\_back();

        return powmod(superPow(a, b), 10) \* powmod(a, last\_digit) % base;

    }

};



75.[Sort Colors](https://leetcode.com/problems/sort-colors/description/)

class Solution {

public:

    void sortColors(vector<int>& nums) {

        int low = 0, mid = 0, high = nums.size()-1;

        while(mid <= high){

            if(nums[mid] == 0){

                swap(nums[low], nums[mid]);

                low++;

                mid++;

            }

            else if(nums[mid] == 1){

                mid++;

            }

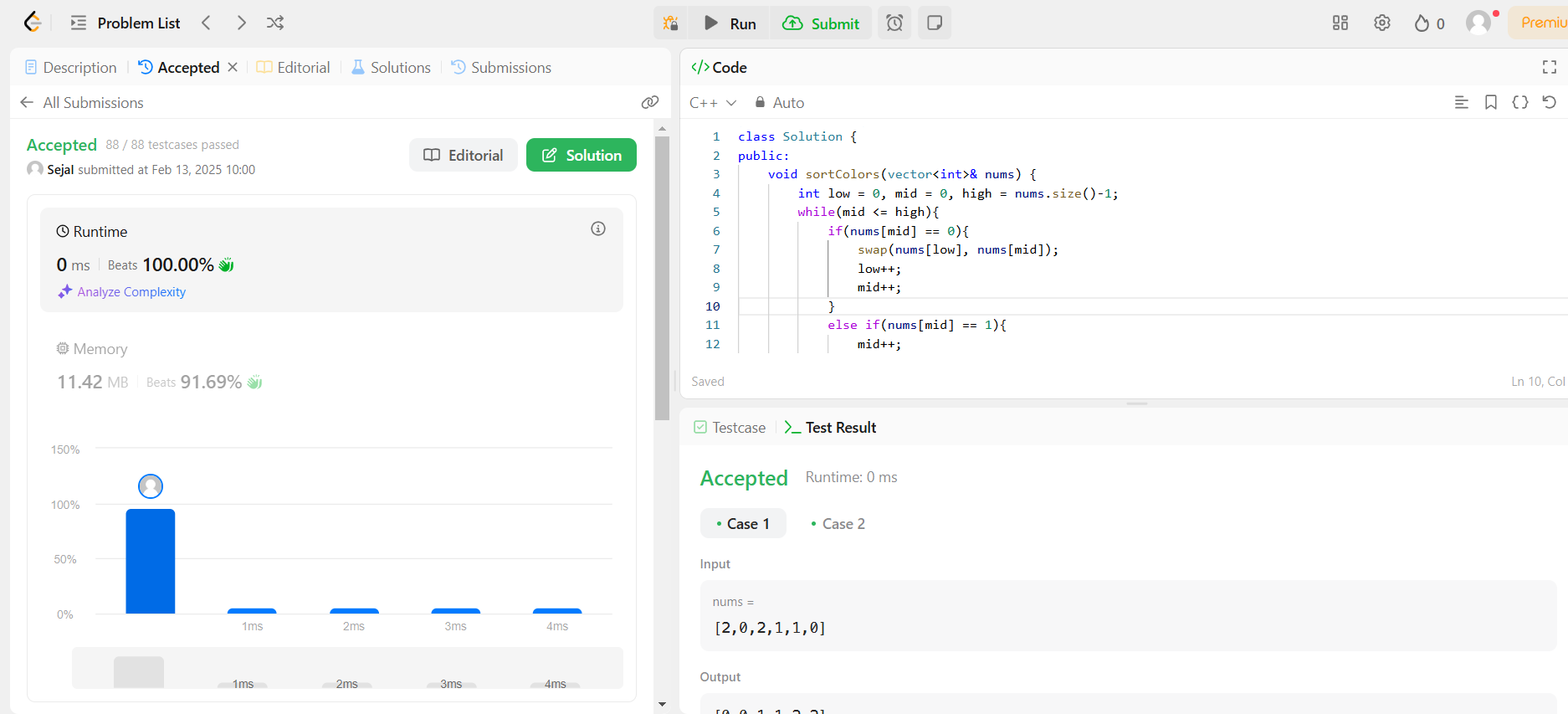
            else{

                swap(nums[mid], nums[high]);

                high--;

            }}}

};



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

class Solution {

public:

    int maxSubArray(vector<int>& nums) {

        int maxSum = INT\_MIN;

        int currentSum = 0;

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

            currentSum += nums[i];

            if (currentSum > maxSum) {

                maxSum = currentSum;

            }

            if (currentSum < 0) {

                currentSum = 0;

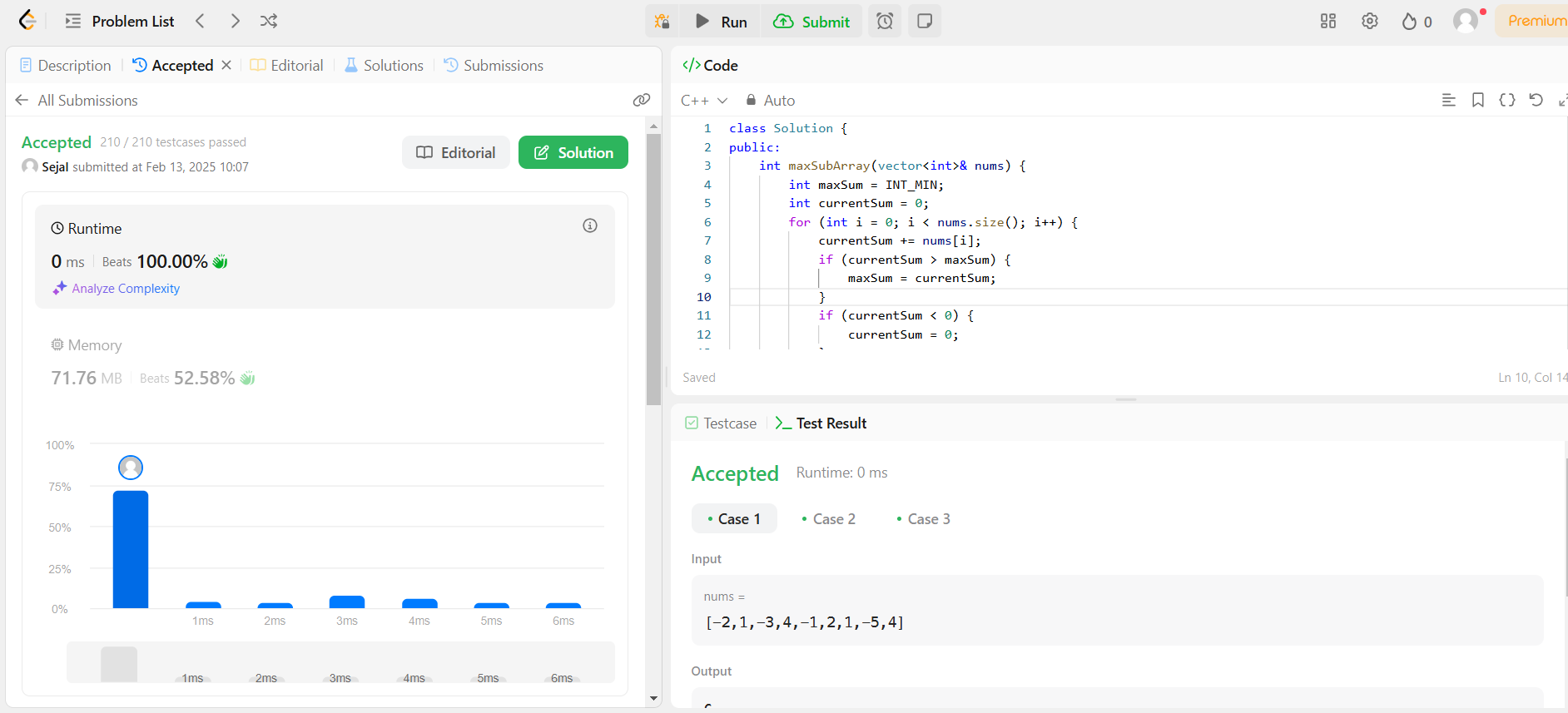
            }

        }

        return maxSum;

    }

};



278.[First Bad Version](https://leetcode.com/problems/first-bad-version/description/)

class Solution {

public:

    int firstBadVersion(int n) {

        int first = 1;

        int last = n;

        while (first < last) {

            int mid = first + (last - first) / 2;

            if (isBadVersion(mid)) {

                last = mid;

            } else {

                first = mid + 1;

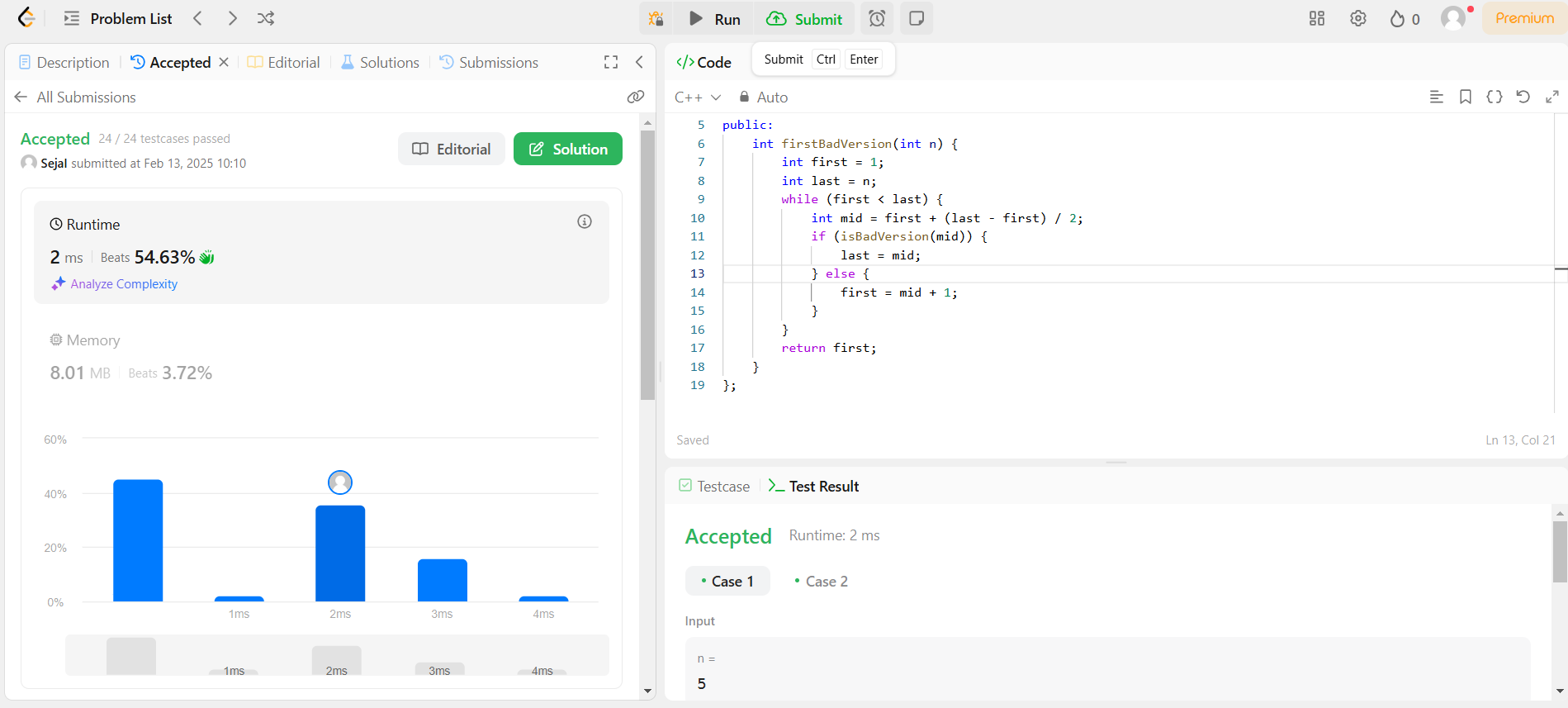
            }

        }

        return first;

    }

};



347.[Top K Frequent Elements](https://leetcode.com/problems/top-k-frequent-elements/description/)

class Solution {

public:

    vector<int> topKFrequent(vector<int>& nums, int k) {

        unordered\_map<int, int> ump;

        for(int i: nums){

            ump[i]++;

        }

        priority\_queue<pair<int, int>>pq;

        for(auto i: ump){

            pq.push({i.second,i.first});

        }

        vector<int> res;

        while(k--){

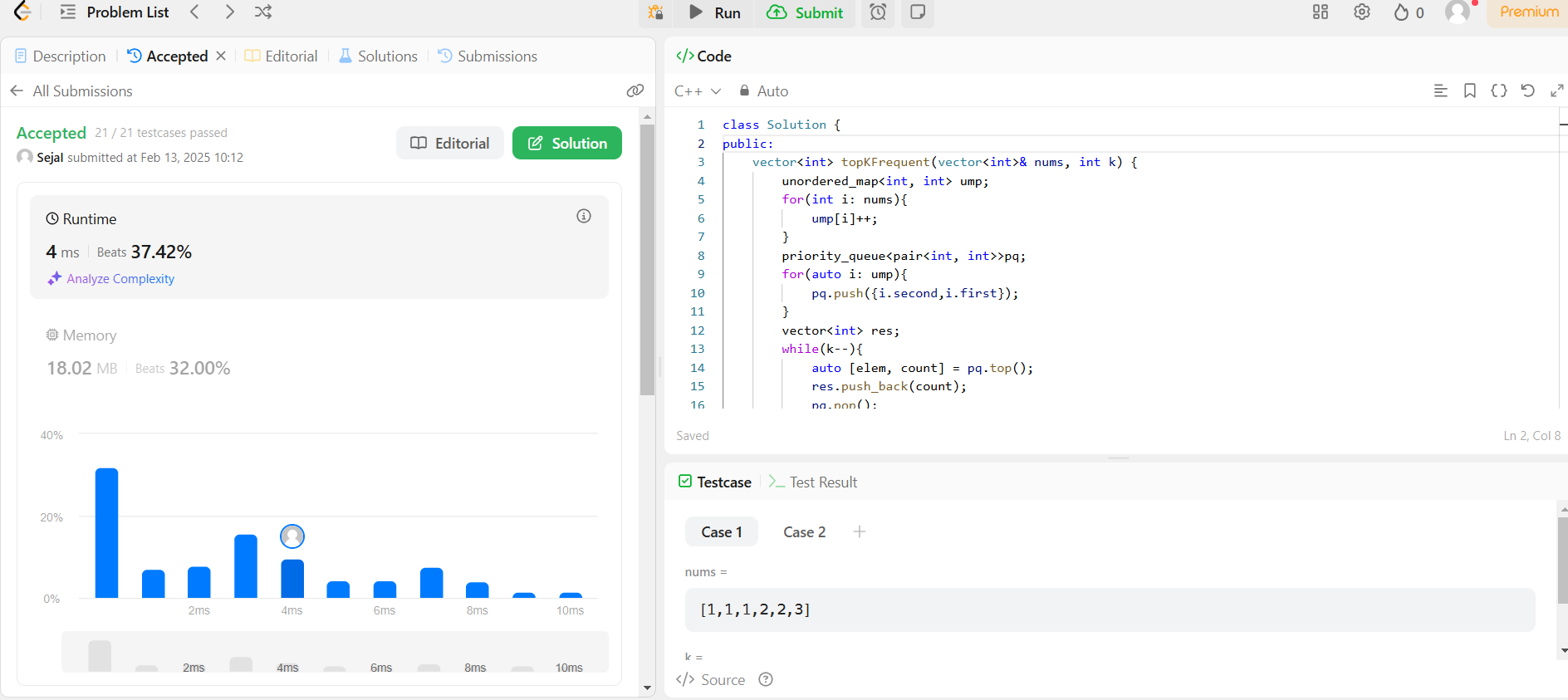
            auto [elem, count] = pq.top();

            res.push\_back(count);

            pq.pop(); }

        return res;

    } };



932.[Beautiful Array](https://leetcode.com/problems/beautiful-array/description/)

class Solution {

public:

int partition(vector<int> &v, int start, int end, int mask)

    {

        int j = start;

        for(int i = start; i <= end; i++)

        {

            if((v[i] & mask) != 0)

            {

                swap(v[i], v[j]);

                j++;

            }

        }

        return j;

    }

    void sort(vector<int> & v, int start, int end, int mask)

    {

        if(start >= end) return;

        int mid = partition(v, start, end, mask);

        sort(v, start, mid - 1, mask << 1);

        sort(v, mid, end, mask << 1);

    }

    vector<int> beautifulArray(int n) {

        vector<int> ans;

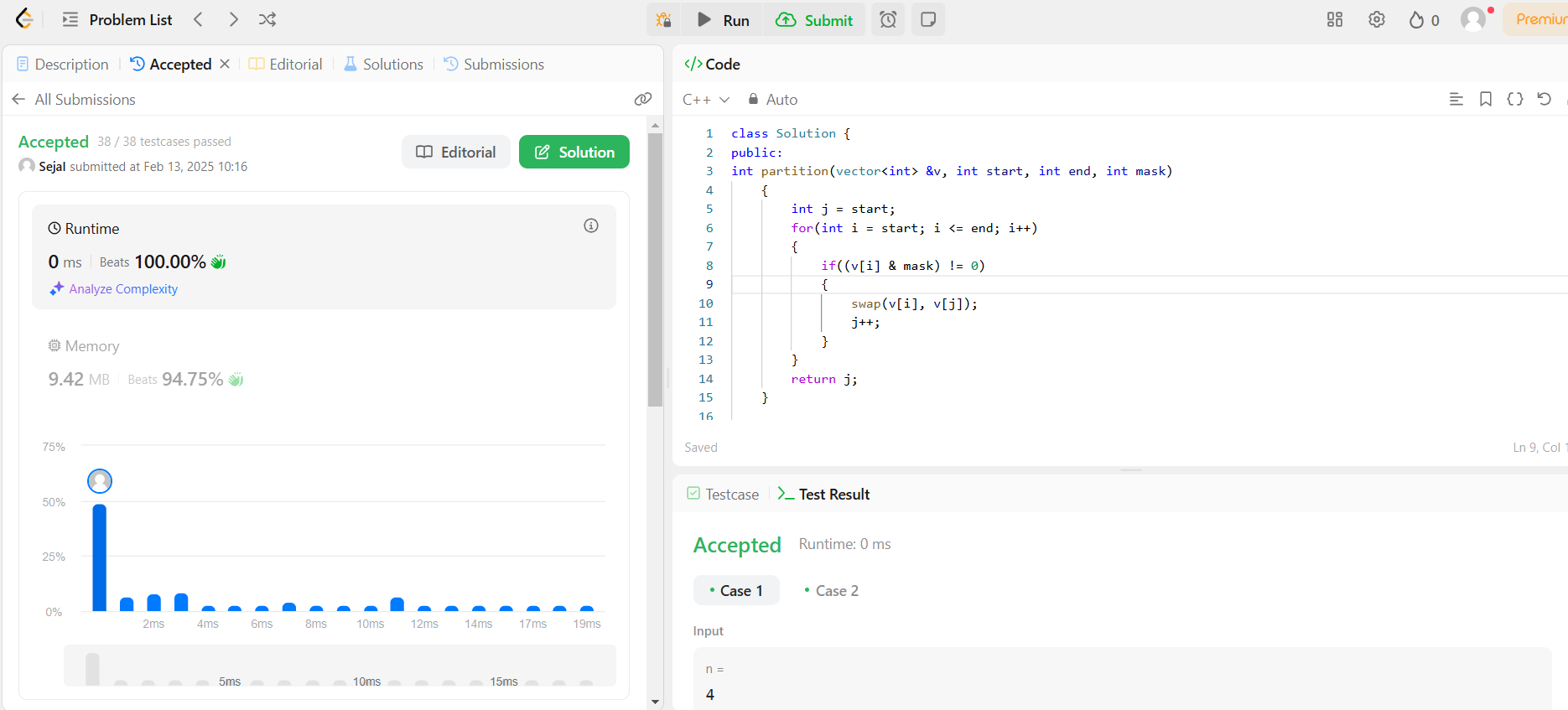
        for(int i = 0; i < n; i++) ans.push\_back(i + 1);

        sort(ans, 0, n - 1, 1);

        return ans;

    }

};



162.[Find Peak Element](https://leetcode.com/problems/find-peak-element/description/)

class Solution {

public:

    int findPeakElement(vector<int>& nums) {

        int n = nums.size();

        int r = n-1;

        int l = 0;

        while(r>l){

            int mid = (r+l)/2;

            if(nums[mid] > nums[mid+1]) r = mid;

            else l = mid +1;

        }

        return l;

    }

};

