UNIVERSITY INSTITUTE OF ENGINEERING

Department of Computer Science & Engineering

(BE-CSE-6th Sem)

Subject Name: Advanced Programming Lab - 2

Subject Code: 22CSP-351

Submitted to: Vishal Sir

Submitted by:

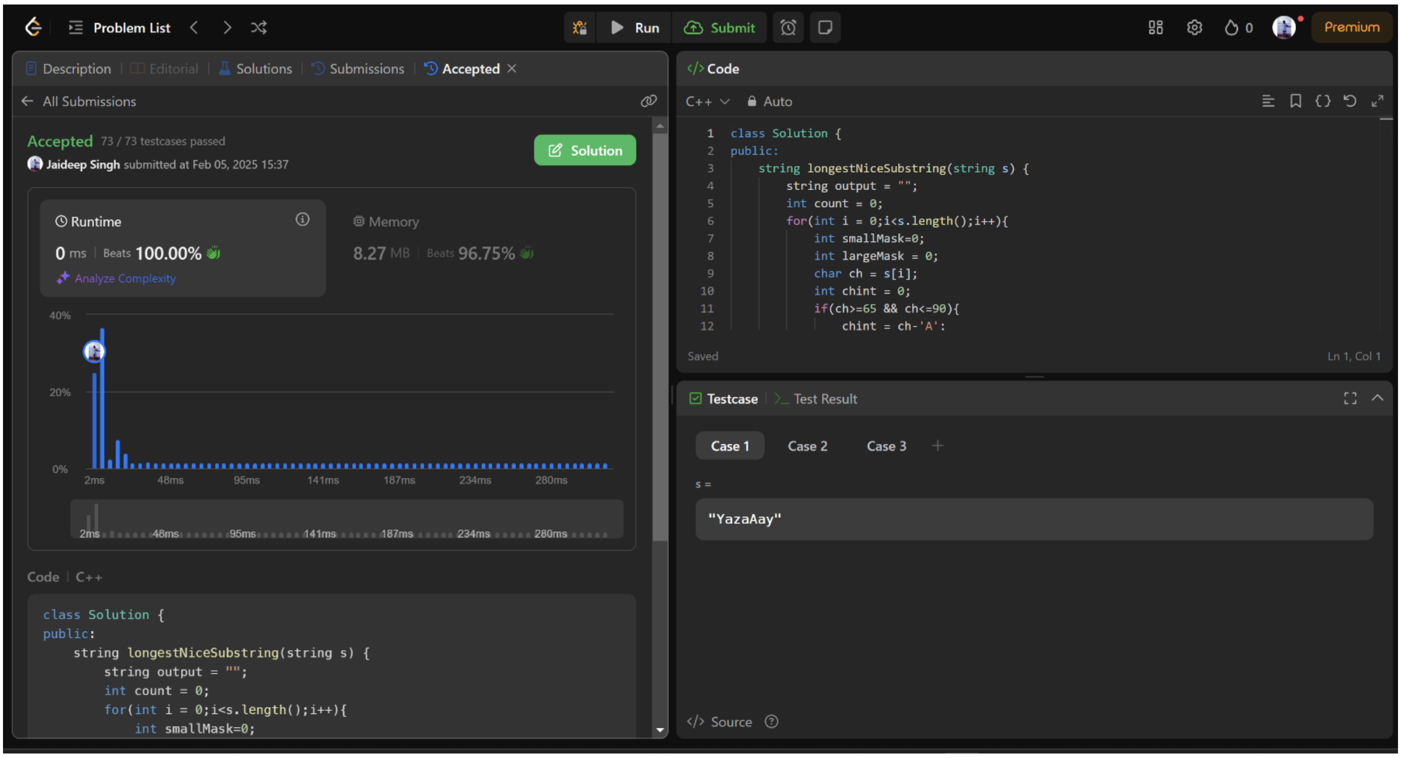
Name: Jaideep Singh

UID: 22BCS16802

Section: FL-IOT-604

Group: A

1763.Longest Nice Substring



Accepted 73 / 73 testcases passed

Jaideep Singh submitted at Feb 05, 2025 15:37

Runtime

0 ms | Beats 100.00%

Memory

8.27 MB | Beats 96.75%

Code | C++

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        string output = "";
        int count = 0;
        for(int i = 0;i<s.length();i++){
            int smallMask=0;
```

Code

C++ Auto

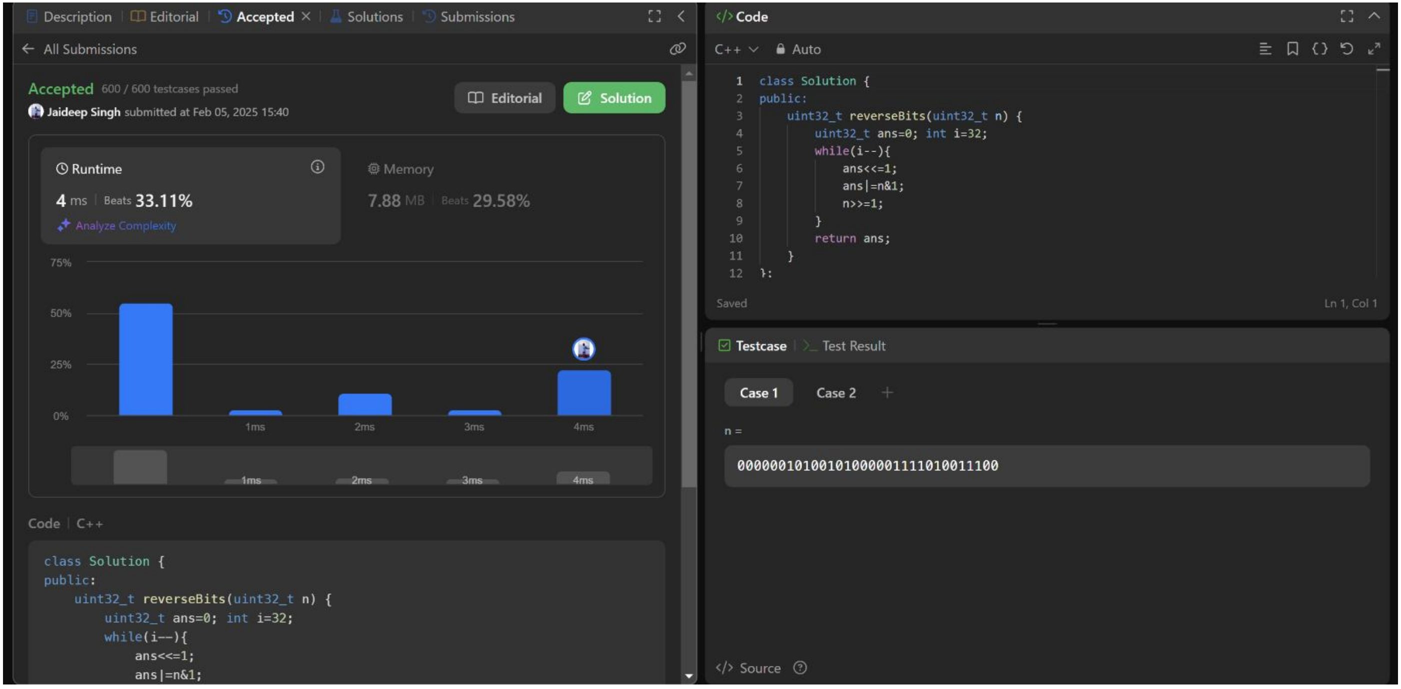
```
class Solution {
public:
    string longestNiceSubstring(string s) {
        string output = "";
        int count = 0;
        for(int i = 0;i<s.length();i++){
            int smallMask=0;
            int largeMask = 0;
            char ch = s[i];
            int chint = 0;
            if(ch>=65 && ch<=90){
                chint = ch-'A':
```

Saved Ln 1, Col 1

s =

```
"YazaAay"
```

190.Reverse Bits



Accepted 600 / 600 testcases passed

Jaideep Singh submitted at Feb 05, 2025 15:40

Runtime

4 ms | Beats 33.11%

Memory

7.88 MB | Beats 29.58%

Code | C++

```
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t ans=0; int i=32;
        while(i--){
            ans<<=1;
            ans|=n&1;
```

Code

C++ Auto

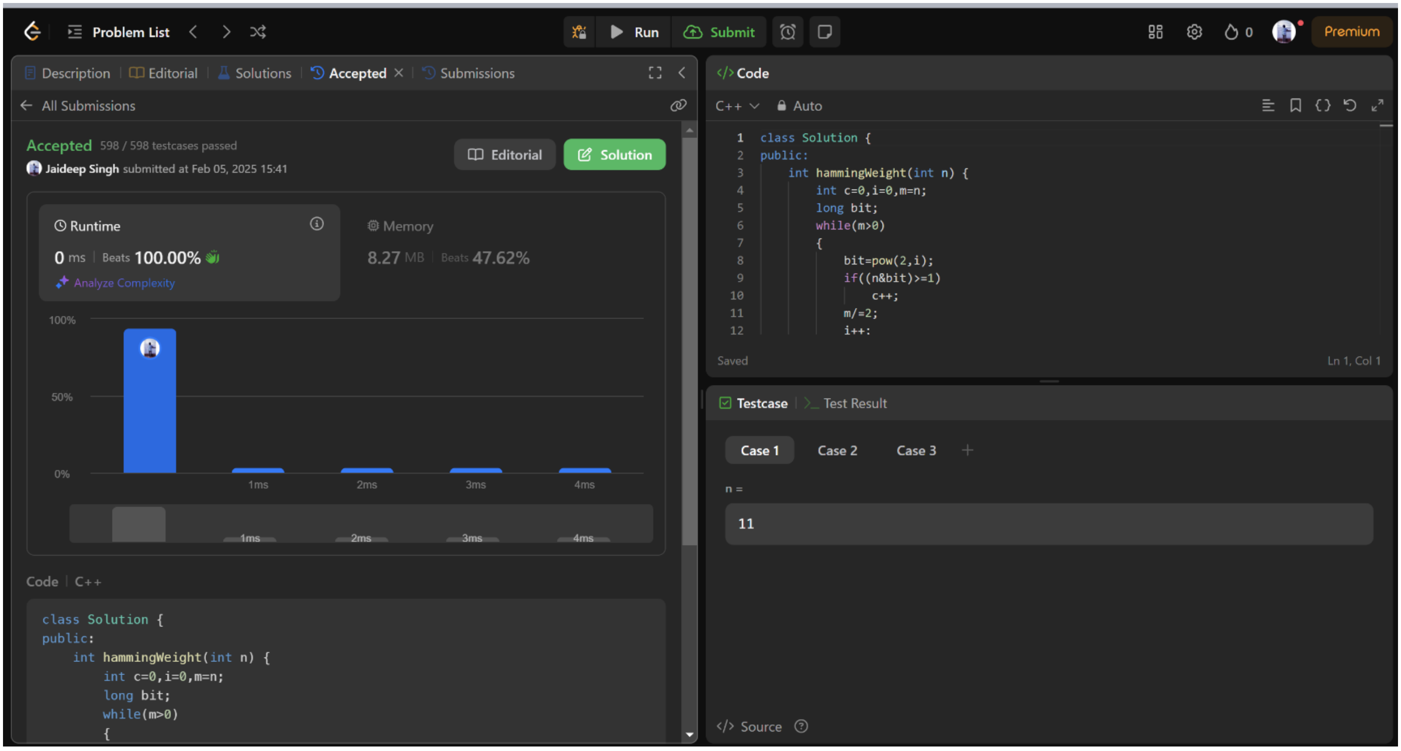
```
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t ans=0; int i=32;
        while(i--){
            ans<<=1;
            ans|=n&1;
            n>>=1;
        }
        return ans;
    }
};
```

Saved Ln 1, Col 1

n =

```
00000010100101000001111010011100
```

191.Number of 1 Bits



Accepted 598 / 598 testcases passed

Jaideep Singh submitted at Feb 05, 2025 15:41

Runtime 0 ms | Beats 100.00%

Memory 8.27 MB | Beats 47.62%

Code | C++

```
class Solution {
public:
    int hammingWeight(int n) {
        int c=0,i=0,m=n;
        long bit;
        while(m>0)
        {
```

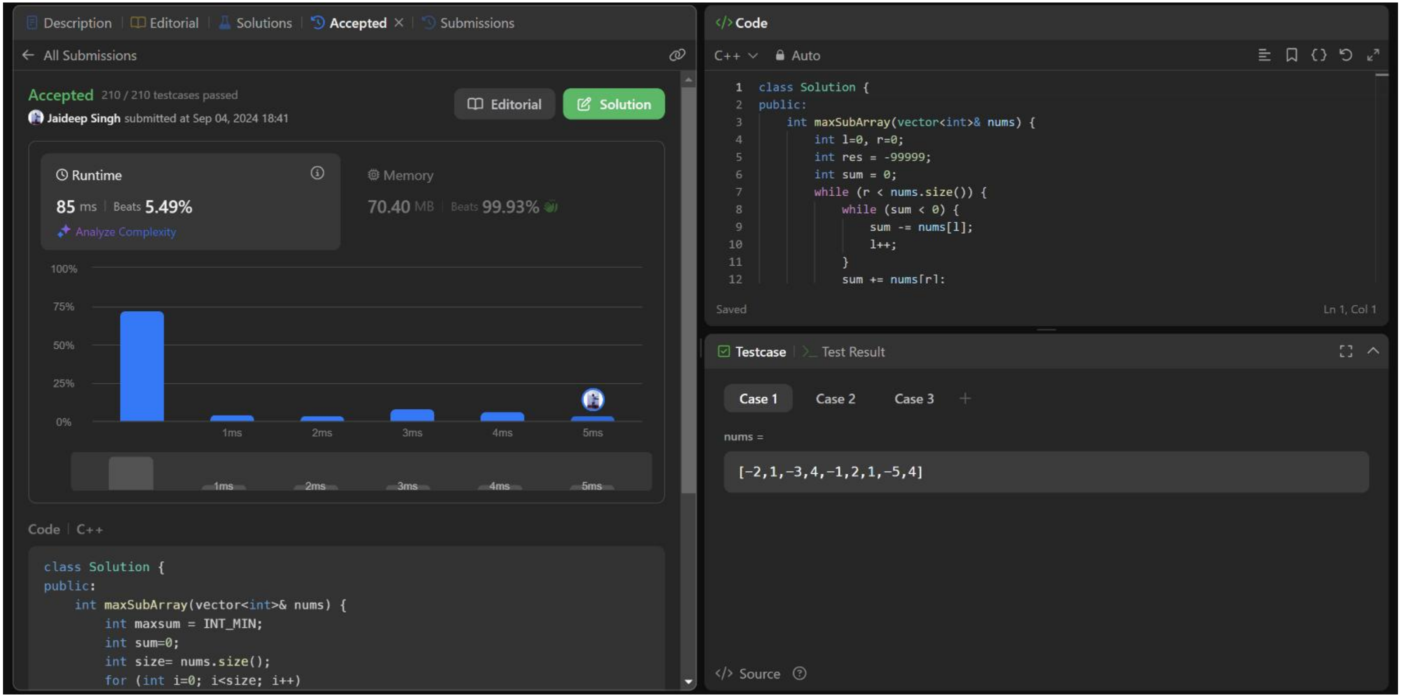
```
class Solution {
public:
    int hammingWeight(int n) {
        int c=0,i=0,m=n;
        long bit;
        while(m>0)
        {
            bit=pow(2,i);
            if((n&bit)>=1)
                c++;
            m/=2;
            i++:
```

Saved Ln 1, Col 1

n =

```
11
```

53.Maximum Subarray



Accepted 210 / 210 testcases passed

Jaideep Singh submitted at Sep 04, 2024 18:41

Runtime 85 ms | Beats 5.49%

Memory 70.40 MB | Beats 99.93%

Code | C++

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int maxsum = INT_MIN;
        int sum=0;
        int size= nums.size();
        for (int i=0; i<size; i++)
```

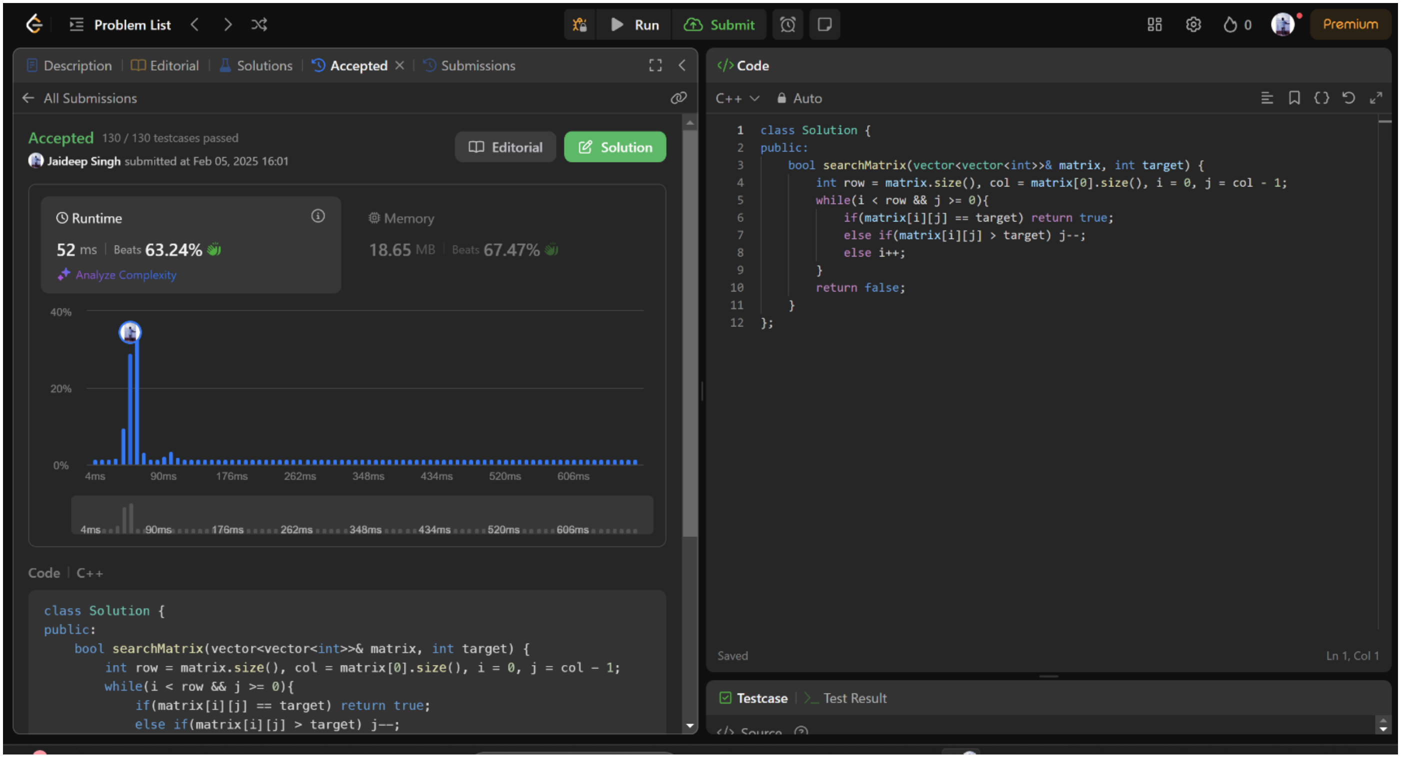
```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int l=0, r=0;
        int res = -99999;
        int sum = 0;
        while (r < nums.size()) {
            while (sum < 0) {
                sum -= nums[l];
                l++;
            }
            sum += nums[r]:
```

Saved Ln 1, Col 1

nums =

```
[-2,1,-3,4,-1,2,1,-5,4]
```

240.Search a 2D Matrix II



Accepted 130 / 130 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:01

Runtime: 52 ms | Beats 63.24%

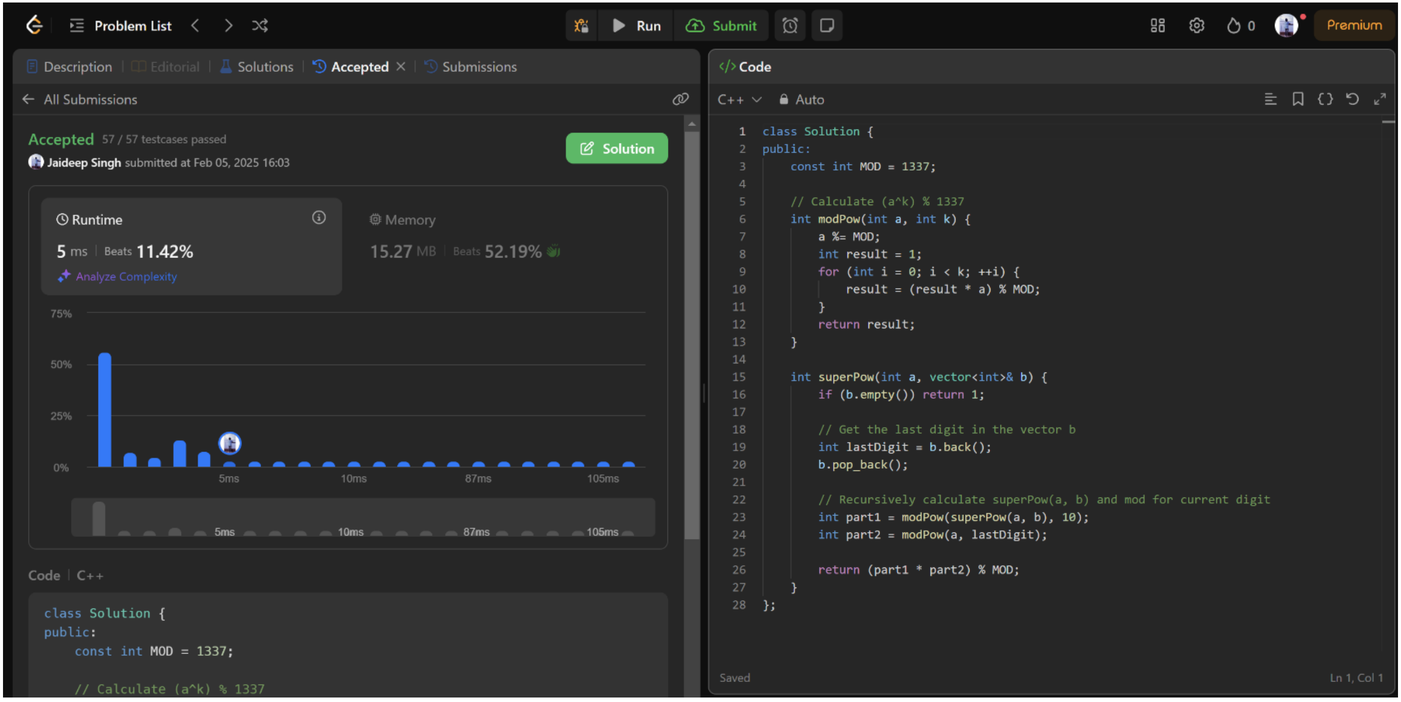
Memory: 18.65 MB | Beats 67.47%

Code | C++

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int row = matrix.size(), col = matrix[0].size(), i = 0, j = col - 1;
        while(i < row && j >= 0){
            if(matrix[i][j] == target) return true;
            else if(matrix[i][j] > target) j--;
```

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int row = matrix.size(), col = matrix[0].size(), i = 0, j = col - 1;
        while(i < row && j >= 0){
            if(matrix[i][j] == target) return true;
            else if(matrix[i][j] > target) j--;
            else i++;
        }
        return false;
    }
};
```

372.Super Pow



Accepted 57 / 57 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:03

Runtime: 5 ms | Beats 11.42%

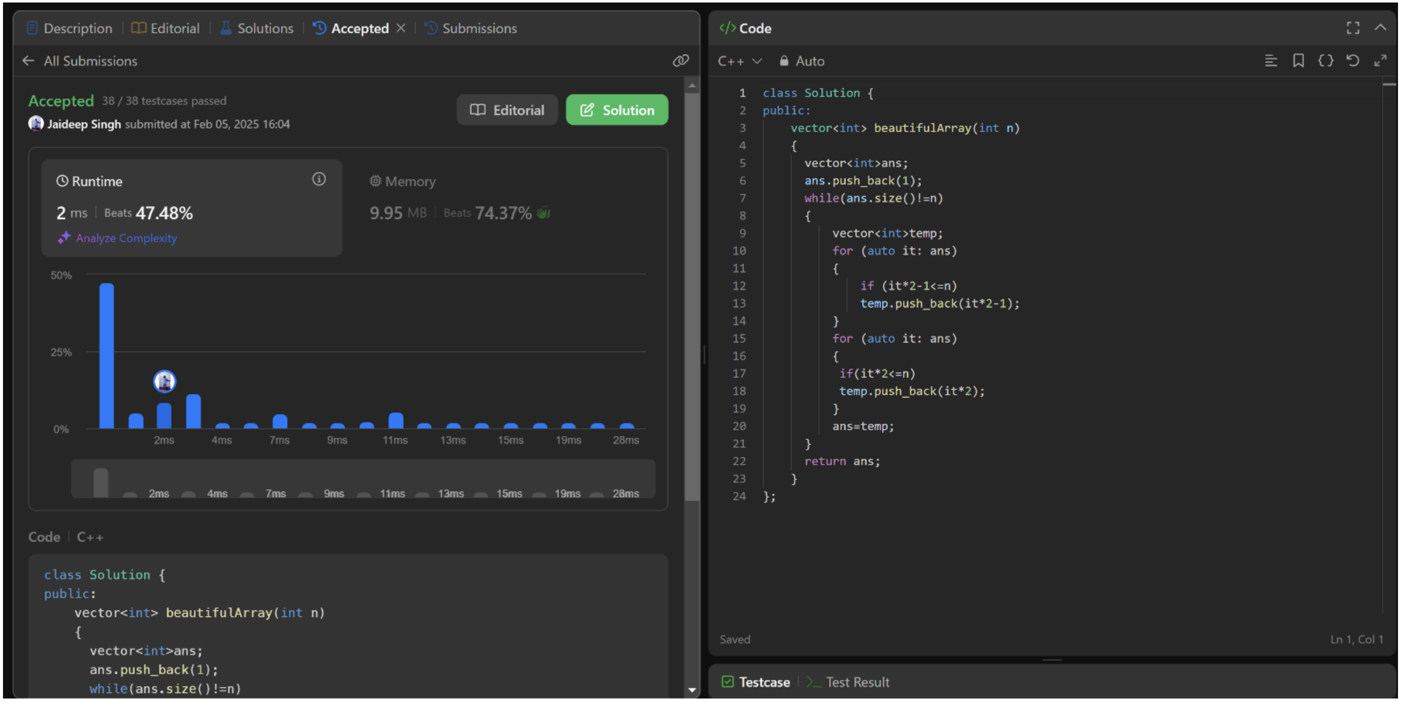
Memory: 15.27 MB | Beats 52.19%

Code | C++

```
class Solution {
public:
    const int MOD = 1337;

    // Calculate (a^k) % 1337
```

```
class Solution {
public:
    const int MOD = 1337;

    // Calculate (a^k) % 1337
    int modPow(int a, int k) {
        a %= MOD;
        int result = 1;
        for (int i = 0; i < k; ++i) {
            result = (result * a) % MOD;
        }
        return result;
    }

    int superPow(int a, vector<int>& b) {
        if (b.empty()) return 1;

        // Get the last digit in the vector b
        int lastDigit = b.back();
        b.pop_back();

        // Recursively calculate superPow(a, b) and mod for current digit
        int part1 = modPow(superPow(a, b), 10);
        int part2 = modPow(a, lastDigit);

        return (part1 * part2) % MOD;
    }
};
```

932.Beautiful Array



Accepted 38 / 38 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:04

Runtime: 2 ms | Beats 47.48%

Memory: 9.95 MB | Beats 74.37%

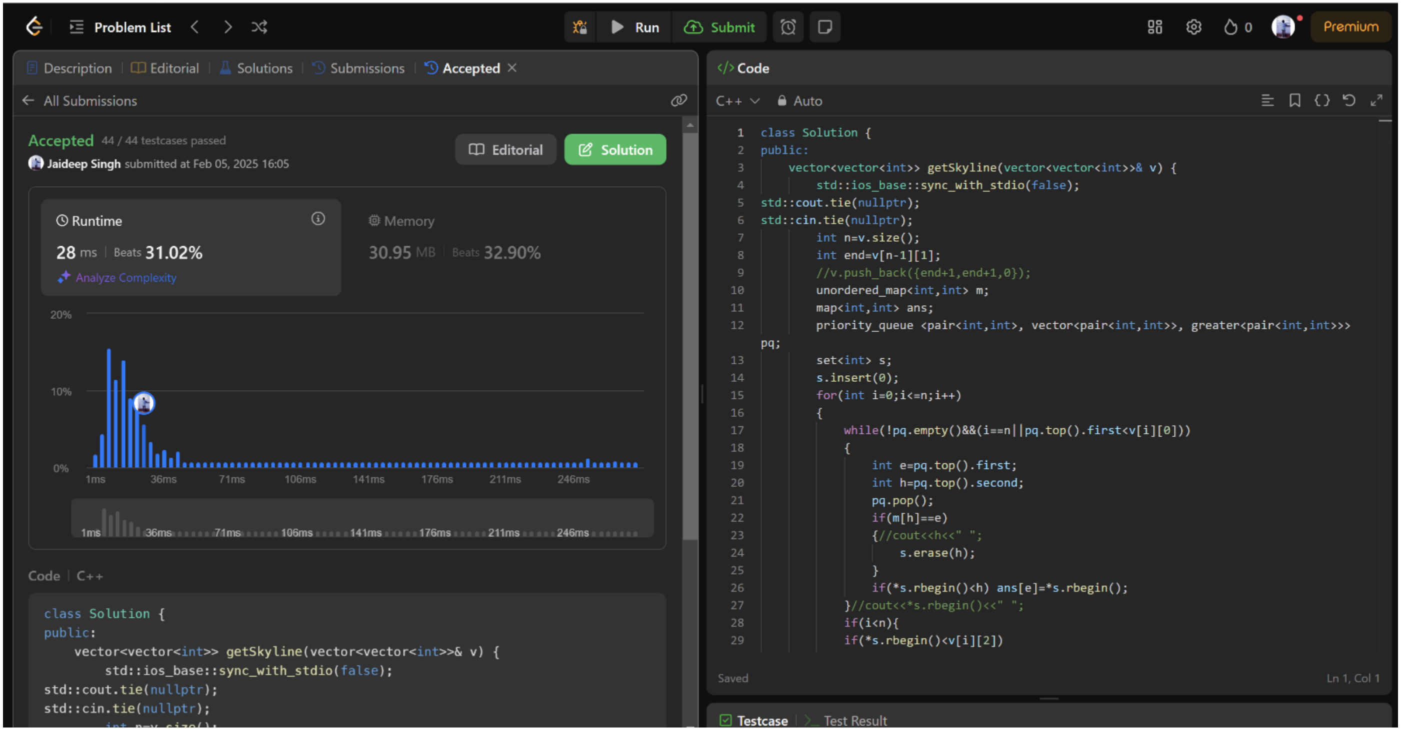
Code | C++

```cpp
class Solution {
public:
    vector<int> beautifulArray(int n)
    {
        vector<int>ans;
        ans.push_back(1);
        while(ans.size()!=n)
        {
            vector<int>temp;
            for (auto it: ans)
            {
                if (it*2-1<=n)
                temp.push_back(it*2-1);
            }
            for (auto it: ans)
            {
             if(it*2<=n)
             temp.push_back(it*2);
            }
            ans=temp;
        }
        return ans;
    }
};
```

Saved Ln 1, Col 1

Testcase | Test Result

218.The Skyline Problem



Accepted 44 / 44 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:05

Runtime: 28 ms | Beats 31.02%

Memory: 30.95 MB | Beats 32.90%

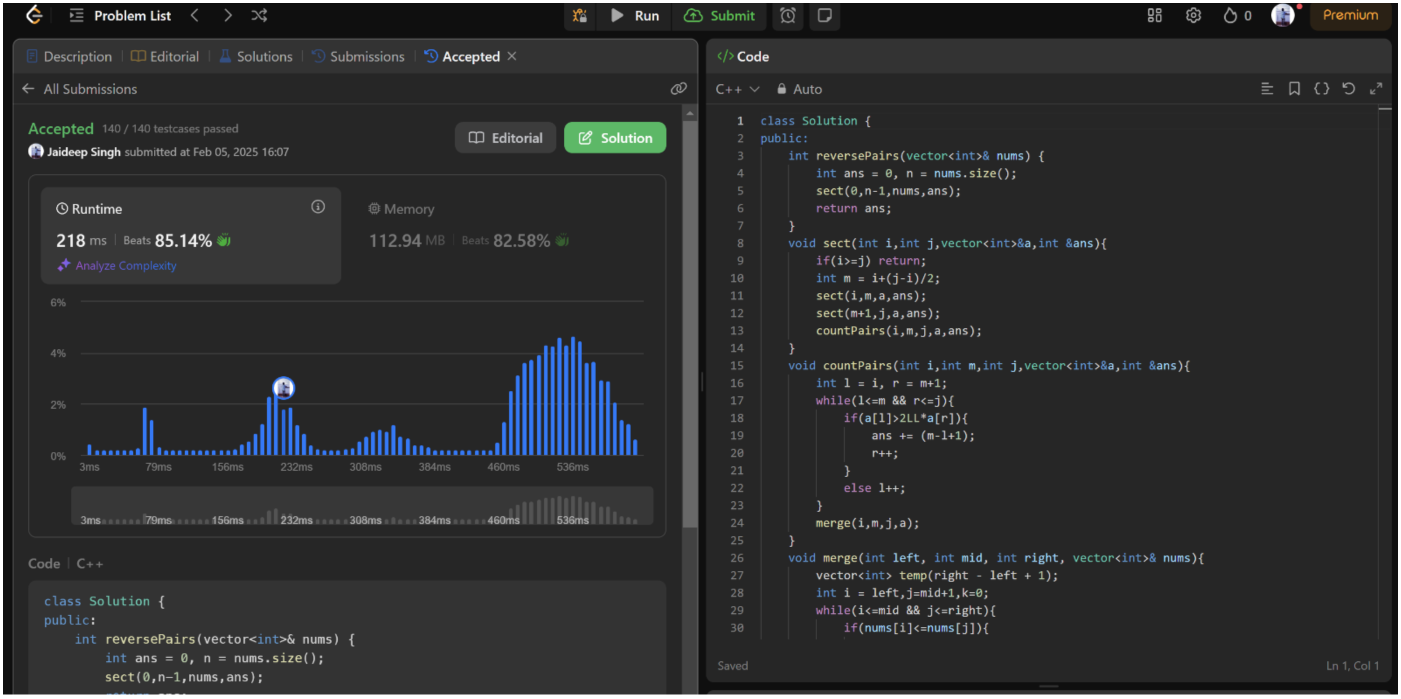
Code | C++

```cpp
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& v) {
        std::ios_base::sync_with_stdio(false);
std::cout.tie(nullptr);
std::cin.tie(nullptr);
        int n=v.size();
        int end=v[n-1][1];
        //v.push_back({end+1,end+1,0});
        unordered_map<int,int> m;
        map<int,int> ans;
        priority_queue <pair<int,int>, vector<pair<int,int>>, greater<pair<int,int>>>
pq;
        set<int> s;
        s.insert(0);
        for(int i=0;i<=n;i++)
        {
            while(!pq.empty()&&(i==n||pq.top().first<v[i][0]))
            {
                int e=pq.top().first;
                int h=pq.top().second;
                pq.pop();
                if(m[h]==e)
                {//cout<<h<<" ";
                    s.erase(h);
                }
                if(*s.rbegin()<h) ans[e]=*s.rbegin();
            }//cout<<*s.rbegin()<<" ";
            if(i<n){
            if(*s.rbegin()<v[i][2])
```

Saved Ln 1, Col 1

Testcase | Test Result

493.Reverse Pairs



Accepted 140 / 140 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:07

Runtime

218 ms | Beats 85.14%

Analyze Complexity

Memory

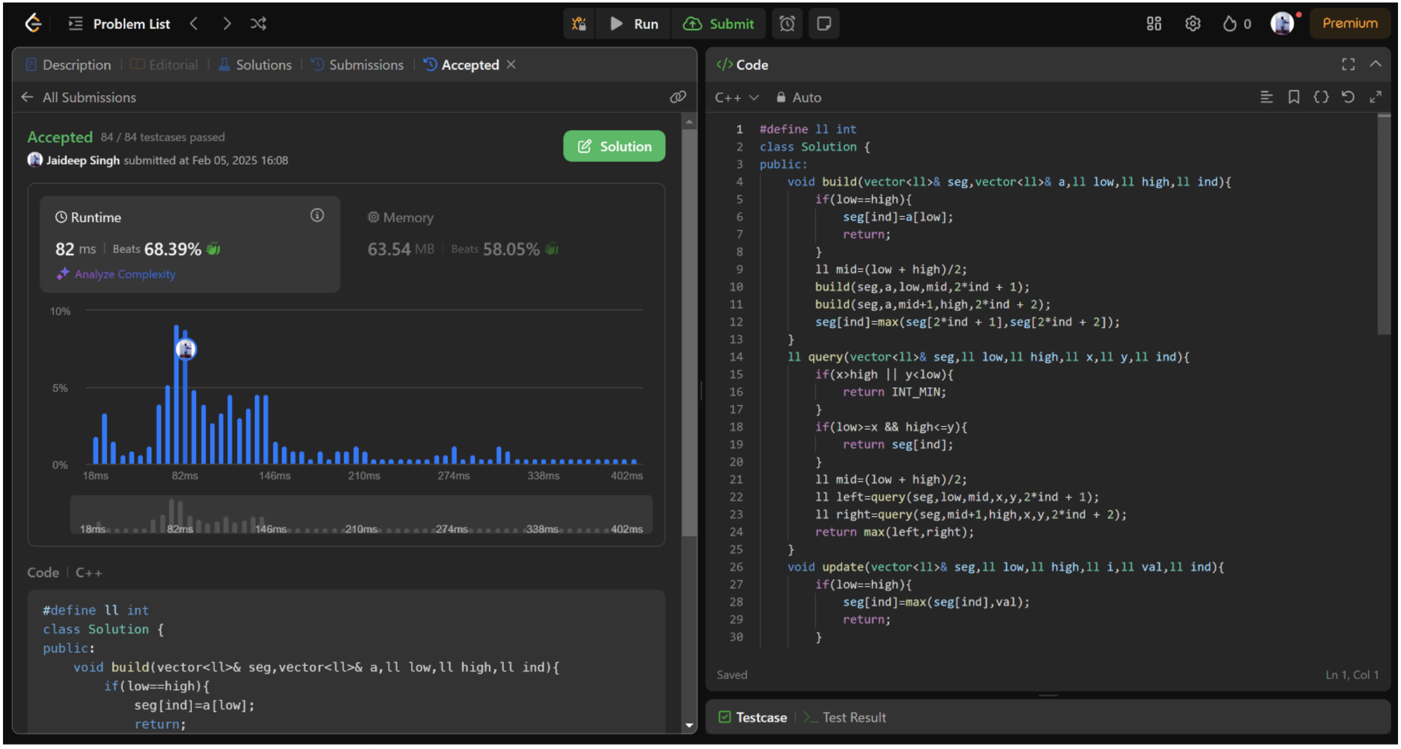
112.94 MB | Beats 82.58%

Code | C++

```
class Solution {
public:
    int reversePairs(vector<int>& nums) {
        int ans = 0, n = nums.size();
        sect(0,n-1,nums,ans);
```

```
class Solution {
public:
    int reversePairs(vector<int>& nums) {
        int ans = 0, n = nums.size();
        sect(0,n-1,nums,ans);
        return ans;
    }
    void sect(int i,int j,vector<int>&a,int &ans){
        if(i>=j) return;
        int m = i+(j-i)/2;
        sect(i,m,a,ans);
        sect(m+1,j,a,ans);
        countPairs(i,m,j,a,ans);
    }
    void countPairs(int i,int m,int j,vector<int>&a,int &ans){
        int l = i, r = m+1;
        while(l<=m && r<=j){
            if(a[l]>2LL*a[r]){
                ans += (m-l+1);
                r++;
            }
            else l++;
        }
        merge(i,m,j,a);
    }
    void merge(int left, int mid, int right, vector<int>& nums){
        vector<int> temp(right - left + 1);
        int i = left,j=mid+1,k=0;
        while(i<=mid && j<=right){
            if(nums[i]<=nums[j]){
```

2407.Longest Increasing Subsequence II



Accepted 84 / 84 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:08

Runtime

82 ms | Beats 68.39%

Analyze Complexity

Memory

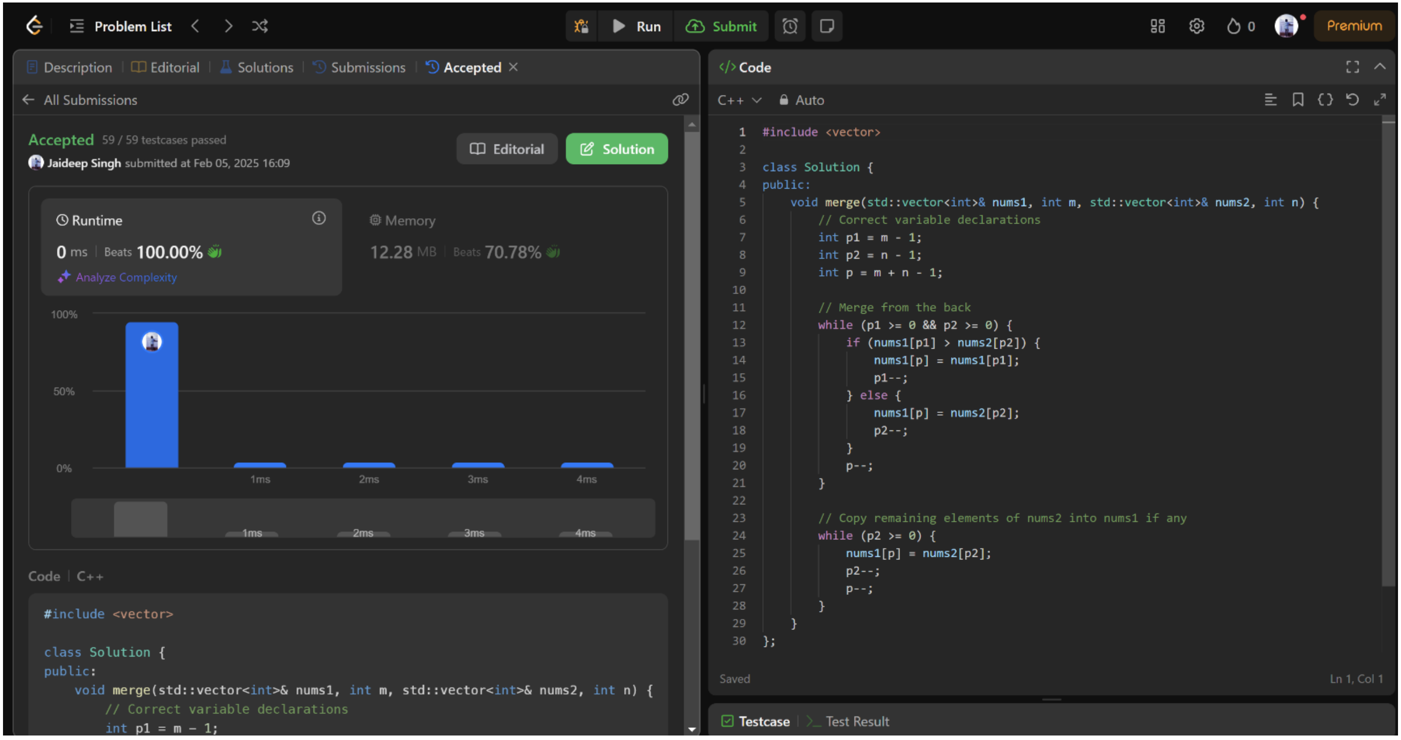
63.54 MB | Beats 58.05%

Code | C++

```
#define ll int
class Solution {
public:
    void build(vector<ll>& seg,vector<ll>& a,ll low,ll high,ll ind){
        if(low==high){
            seg[ind]=a[low];
            return;
```

```
#define ll int
class Solution {
public:
    void build(vector<ll>& seg,vector<ll>& a,ll low,ll high,ll ind){
        if(low==high){
            seg[ind]=a[low];
            return;
        }
        ll mid=(low + high)/2;
        build(seg,a,low,mid,2*ind + 1);
        build(seg,a,mid+1,high,2*ind + 2);
        seg[ind]=max(seg[2*ind + 1],seg[2*ind + 2]);
    }
    ll query(vector<ll>& seg,ll low,ll high,ll x,ll y,ll ind){
        if(x>high || y<low){
            return INT_MIN;
        }
        if(low>=x && high<=y){
            return seg[ind];
        }
        ll mid=(low + high)/2;
        ll left=query(seg,low,mid,x,y,2*ind + 1);
        ll right=query(seg,mid+1,high,x,y,2*ind + 2);
        return max(left,right);
    }
    void update(vector<ll>& seg,ll low,ll high,ll i,ll val,ll ind){
        if(low==high){
            seg[ind]=max(seg[ind],val);
            return;
        }
```

88.Merge Sorted Array



Accepted 59 / 59 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:09

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

12.28 MB | Beats 70.78%

Code | C++

```
#include <vector>

class Solution {
public:
    void merge(std::vector<int>& nums1, int m, std::vector<int>& nums2, int n) {
        // Correct variable declarations
        int p1 = m - 1;
```

Code

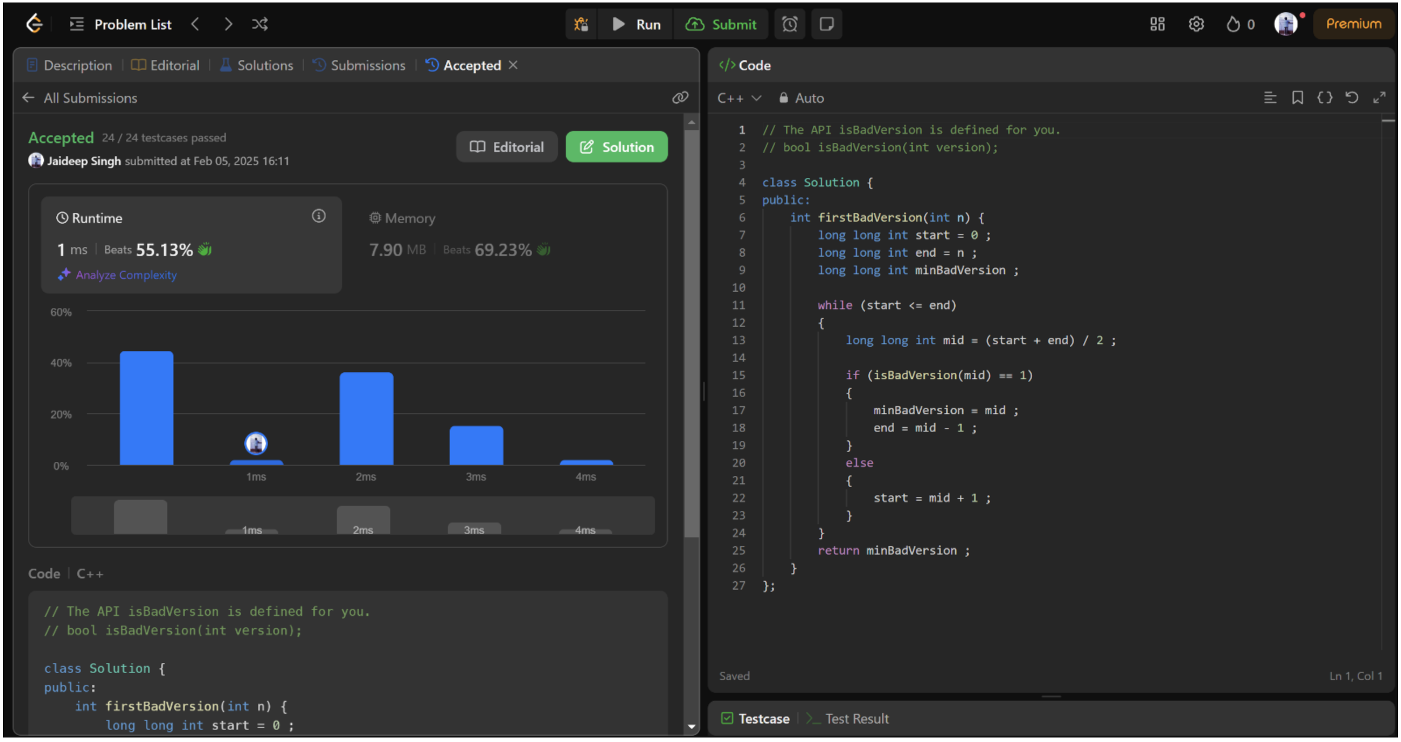
C++ Auto

```
#include <vector>

class Solution {
public:
    void merge(std::vector<int>& nums1, int m, std::vector<int>& nums2, int n) {
        // Correct variable declarations
        int p1 = m - 1;
        int p2 = n - 1;
        int p = m + n - 1;

        // Merge from the back
        while (p1 >= 0 && p2 >= 0) {
            if (nums1[p1] > nums2[p2]) {
                nums1[p] = nums1[p1];
                p1--;
            } else {
                nums1[p] = nums2[p2];
                p2--;
            }
            p--;
        }

        // Copy remaining elements of nums2 into nums1 if any
        while (p2 >= 0) {
            nums1[p] = nums2[p2];
            p2--;
            p--;
        }
    }
};
```

Saved Ln 1, Col 1

Testcase | Test Result

278.First Bad Version



Accepted 24 / 24 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:11

Runtime

1 ms | Beats 55.13%

Analyze Complexity

Memory

7.90 MB | Beats 69.23%

Code | C++

```
// The API isBadVersion is defined for you.
// bool isBadVersion(int version);

class Solution {
public:
    int firstBadVersion(int n) {
        long long int start = 0 ;
```

Code

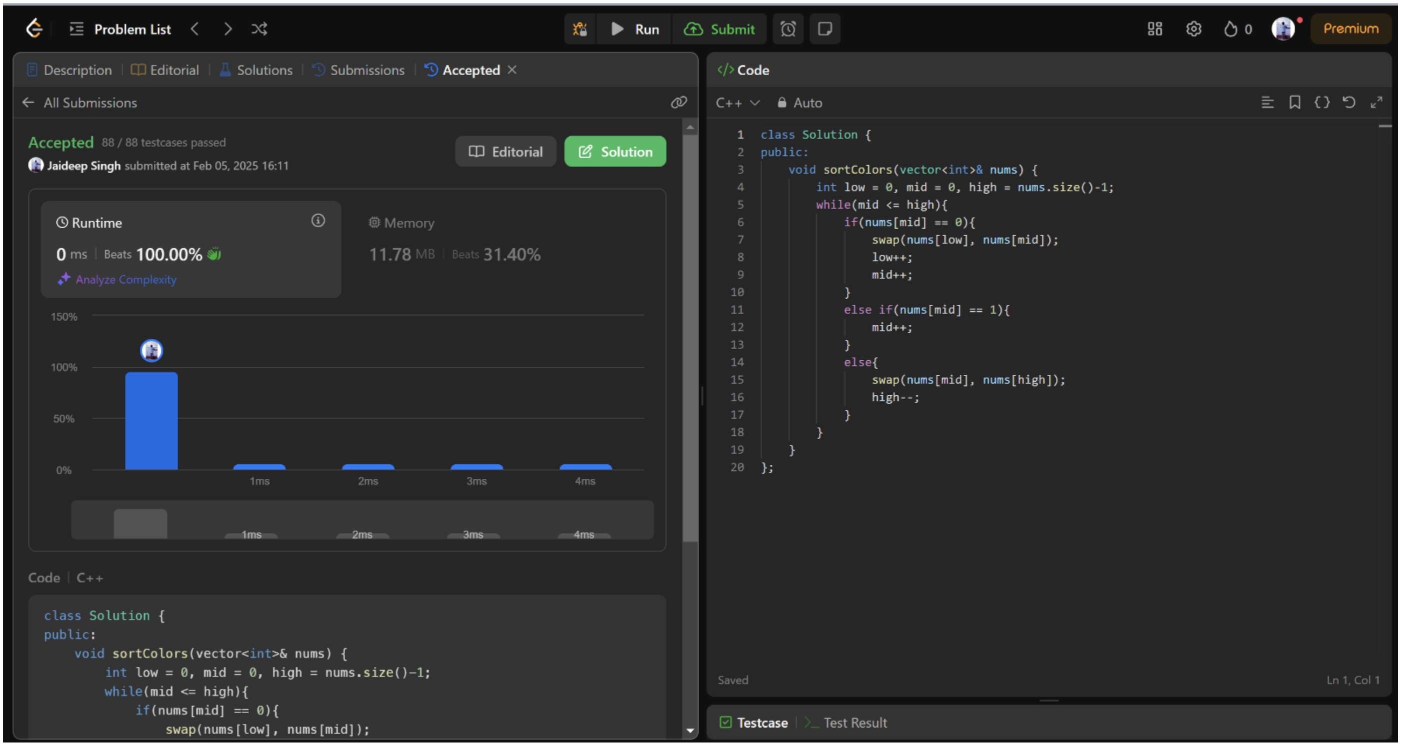
C++ Auto

```
// The API isBadVersion is defined for you.
// bool isBadVersion(int version);

class Solution {
public:
    int firstBadVersion(int n) {
        long long int start = 0 ;
        long long int end = n ;
        long long int minBadVersion ;

        while (start <= end)
        {
            long long int mid = (start + end) / 2 ;

            if (isBadVersion(mid) == 1)
            {
                minBadVersion = mid ;
                end = mid - 1 ;
            }
            else
            {
                start = mid + 1 ;
            }
        }
        return minBadVersion ;
    }
};
```

Saved Ln 1, Col 1

Testcase | Test Result

75.Sort Colors



Accepted 88 / 88 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:11

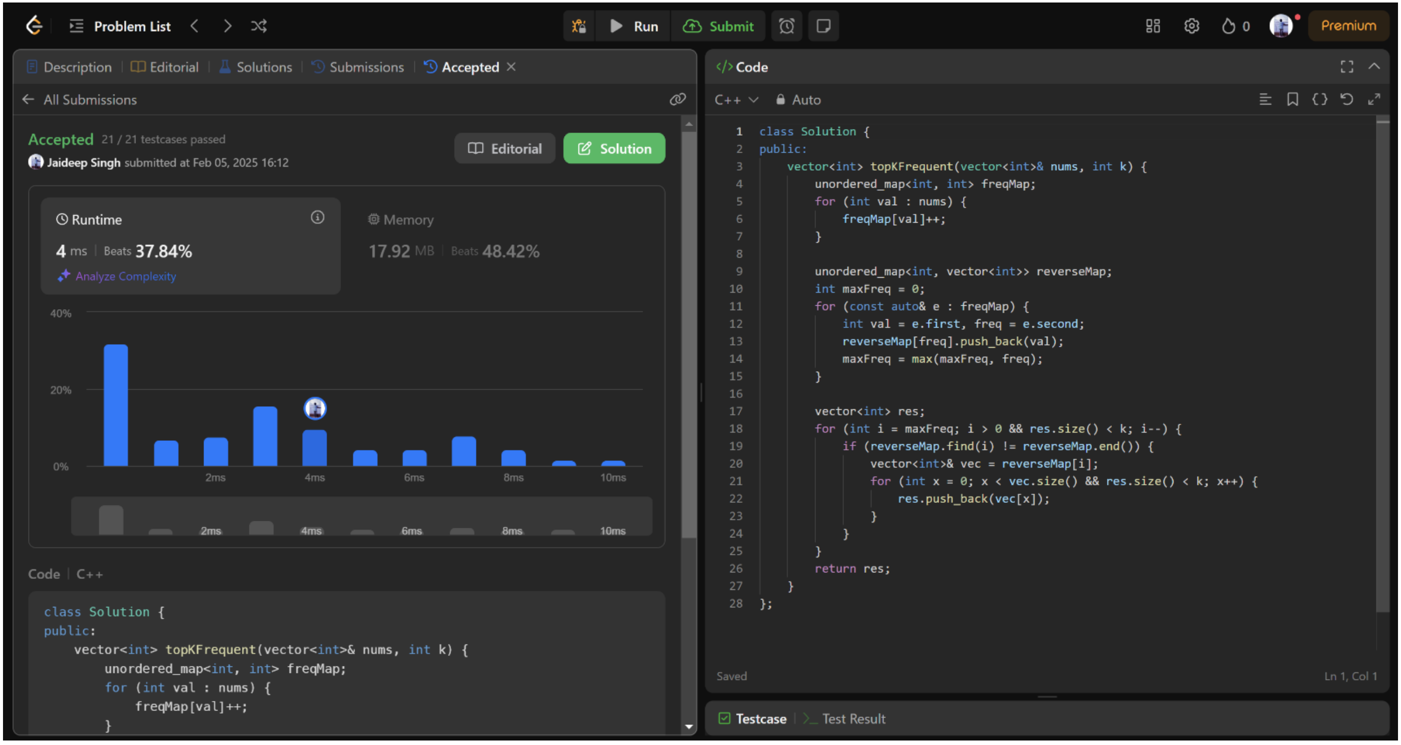
Runtime 0 ms | Beats 100.00%

Memory 11.78 MB | Beats 31.40%

```cpp
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--;
            }
        }
    }
};
```

```cpp
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]);
```

347.Top K Frequent Elements



Accepted 21 / 21 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:12

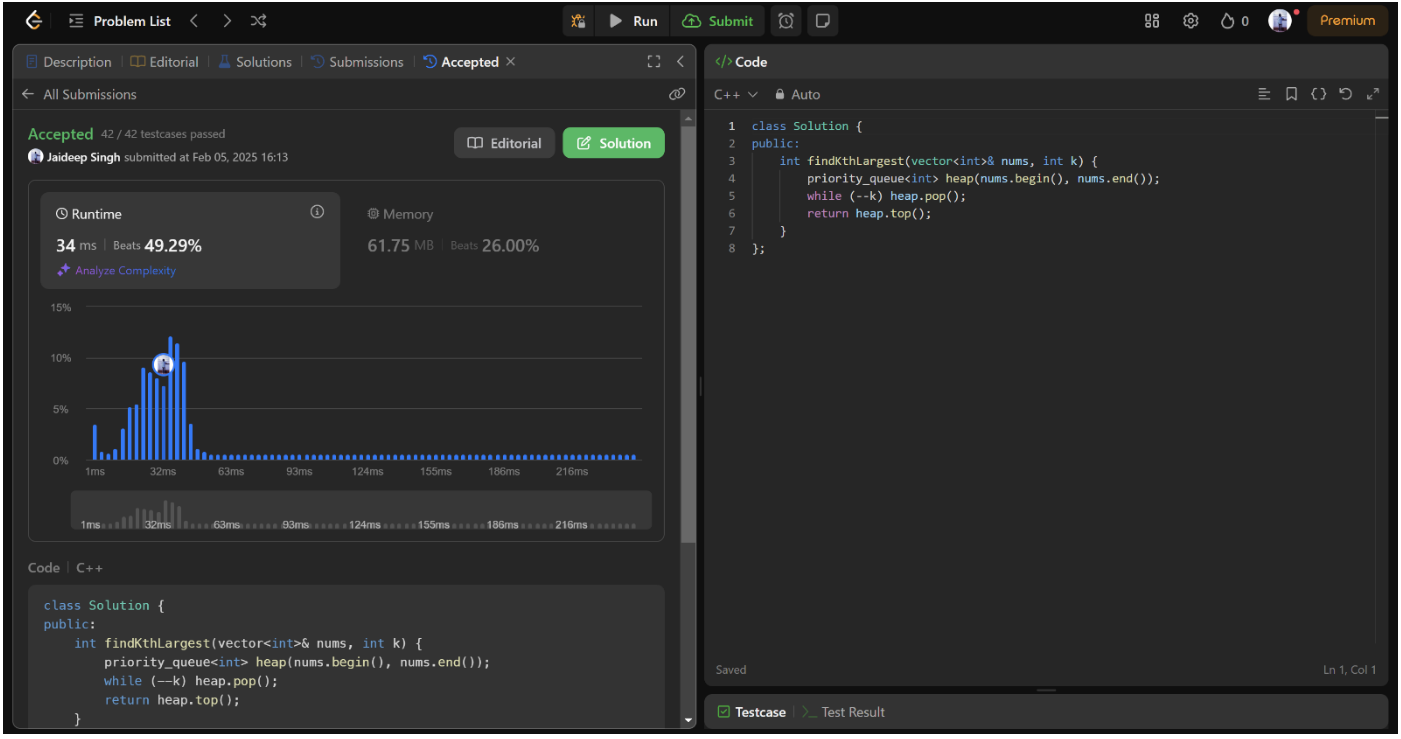
Runtime 4 ms | Beats 37.84%

Memory 17.92 MB | Beats 48.42%

```cpp
class Solution {
public:
    vector<int> topKFrequent(vector<int>& nums, int k) {
        unordered_map<int, int> freqMap;
        for (int val : nums) {
            freqMap[val]++;
        }

        unordered_map<int, vector<int>> reverseMap;
        int maxFreq = 0;
        for (const auto& e : freqMap) {
            int val = e.first, freq = e.second;
            reverseMap[freq].push_back(val);
            maxFreq = max(maxFreq, freq);
        }

        vector<int> res;
        for (int i = maxFreq; i > 0 && res.size() < k; i--) {
            if (reverseMap.find(i) != reverseMap.end()) {
                vector<int>& vec = reverseMap[i];
                for (int x = 0; x < vec.size() && res.size() < k; x++) {
                    res.push_back(vec[x]);
                }
            }
        }
        return res;
    }
};
```

```cpp
class Solution {
public:
    vector<int> topKFrequent(vector<int>& nums, int k) {
        unordered_map<int, int> freqMap;
        for (int val : nums) {
            freqMap[val]++;
        }
```

215.Kth Largest Element in an Array



Accepted 42 / 42 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:13

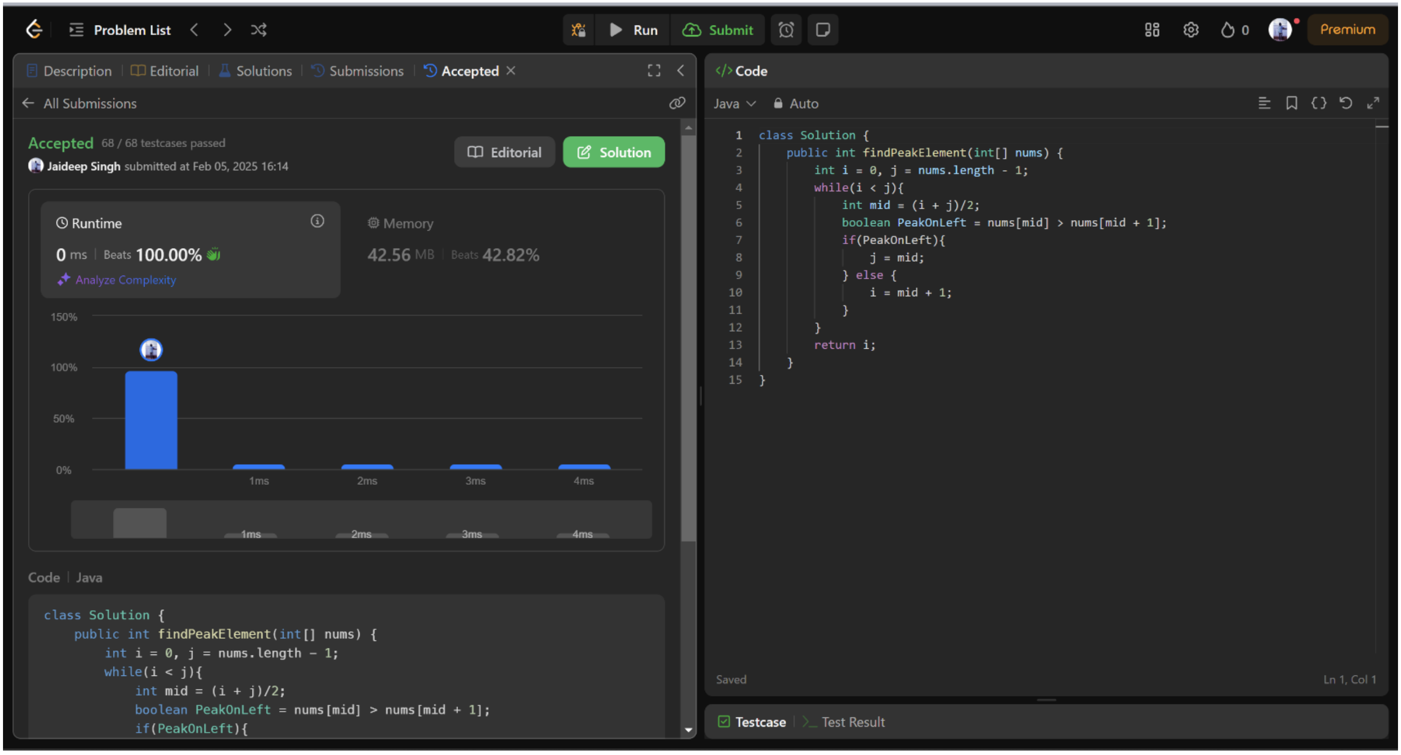
Runtime 34 ms | Beats 49.29%

Memory 61.75 MB | Beats 26.00%

Code | C++

```
class Solution {
public:
    int findKthLargest(vector<int>& nums, int k) {
        priority_queue<int> heap(nums.begin(), nums.end());
        while (--k) heap.pop();
        return heap.top();
    }
};
```

162.Find Peak Element



Accepted 68 / 68 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:14

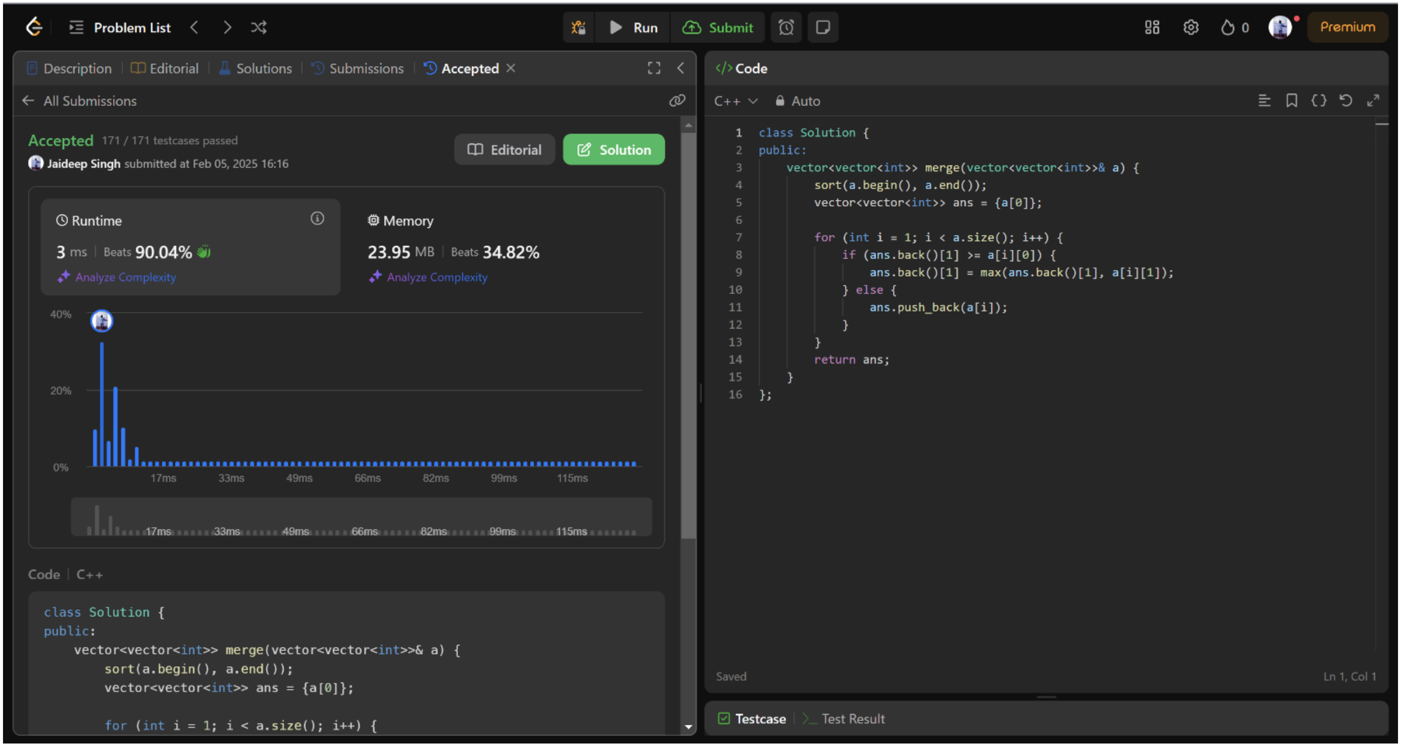
Runtime 0 ms | Beats 100.00%

Memory 42.56 MB | Beats 42.82%

Code | Java

```
class Solution {
    public int findPeakElement(int[] nums) {
        int i = 0, j = nums.length - 1;
        while(i < j){
            int mid = (i + j)/2;
            boolean PeakOnLeft = nums[mid] > nums[mid + 1];
            if(PeakOnLeft){
                j = mid;
            } else {
                i = mid + 1;
            }
        }
        return i;
    }
}
```

56.Merge Intervals



Accepted 171 / 171 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:16

Runtime 3 ms | Beats 90.04%

Memory 23.95 MB | Beats 34.82%

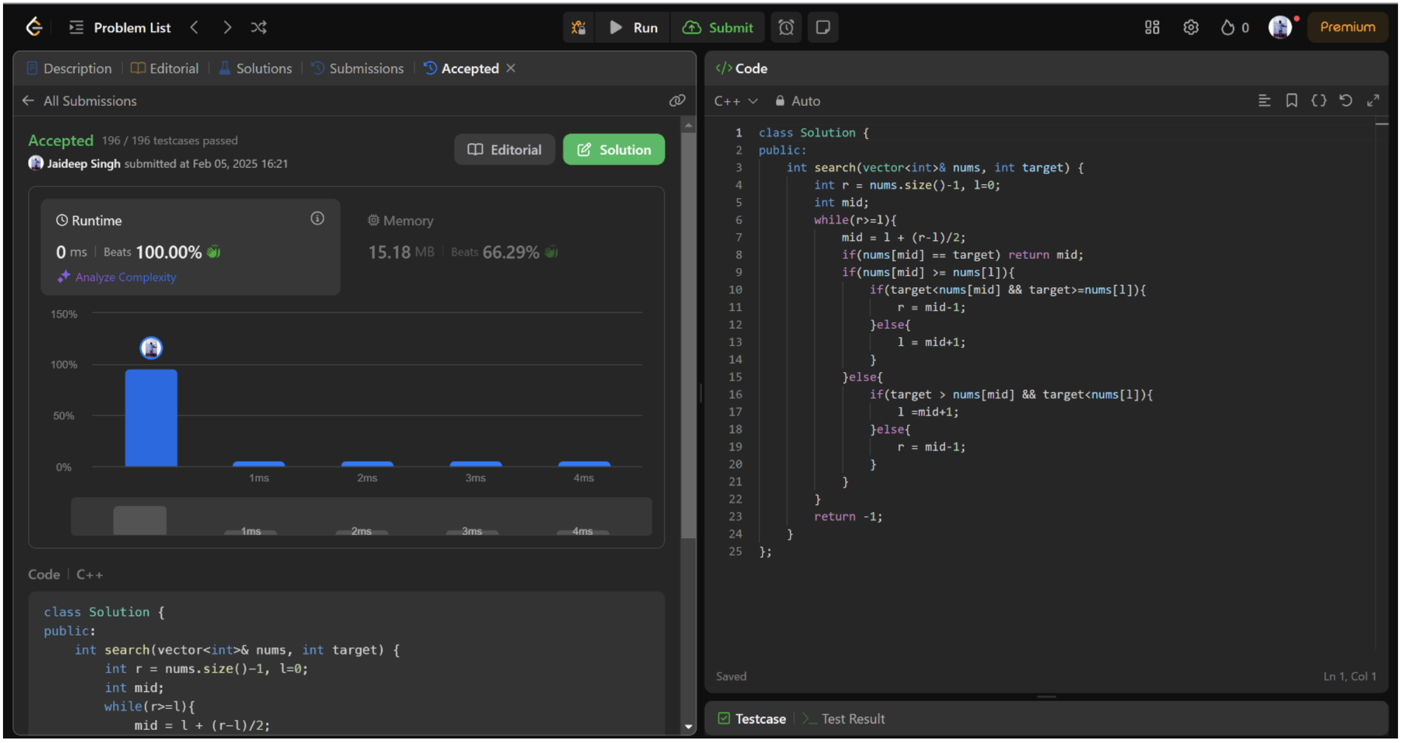
Code | C++

```cpp
class Solution {
public:
    vector<vector<int>> merge(vector<vector<int>>& a) {
        sort(a.begin(), a.end());
        vector<vector<int>> ans = {a[0]};

        for (int i = 1; i < a.size(); i++) {
            if (ans.back()[1] >= a[i][0]) {
                ans.back()[1] = max(ans.back()[1], a[i][1]);
            } else {
                ans.push_back(a[i]);
            }
        }
        return ans;
    }
};
```

class Solution {
public:
    vector<vector<int>> merge(vector<vector<int>>& a) {
        sort(a.begin(), a.end());
        vector<vector<int>> ans = {a[0]};

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

33.Search in Rotated Sorted Array



Accepted 196 / 196 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:21

Runtime 0 ms | Beats 100.00%

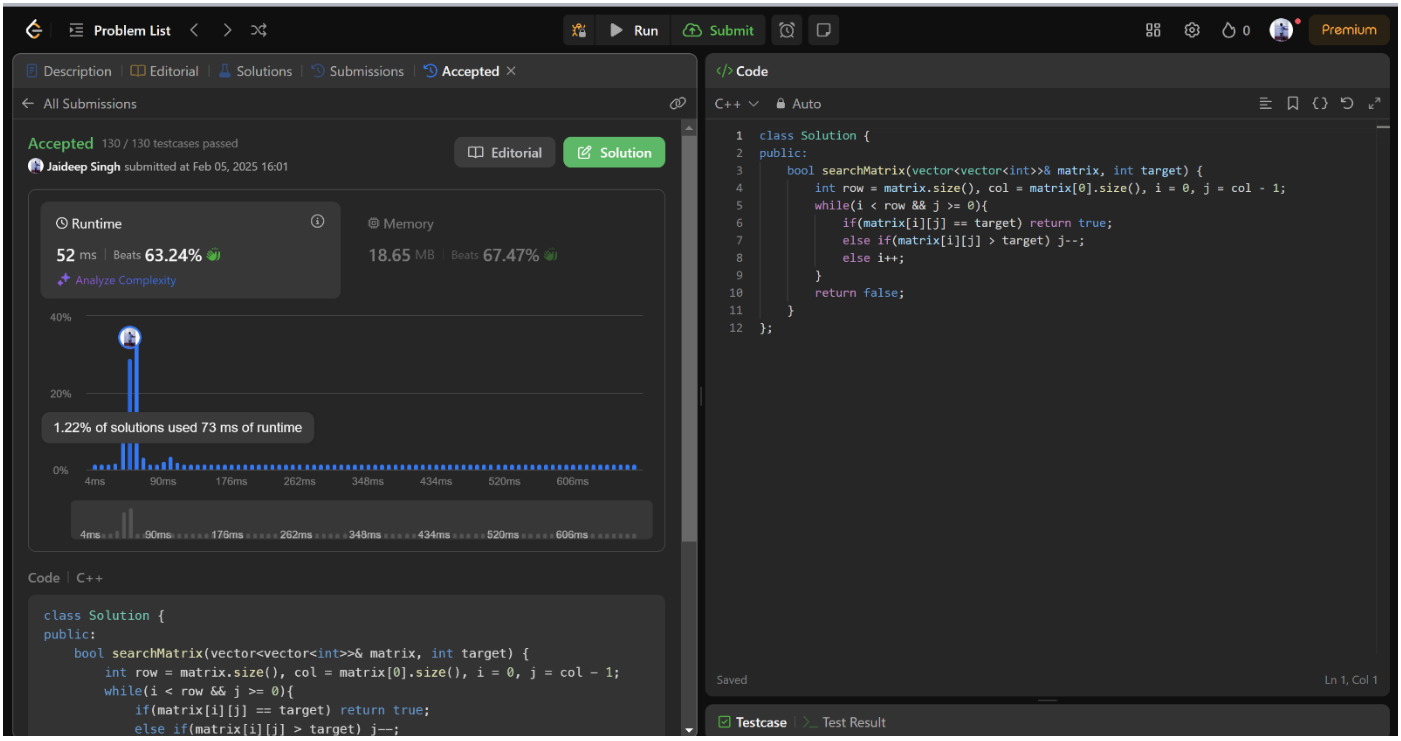
Memory 15.18 MB | Beats 66.29%

Code | C++

```cpp
class Solution {
public:
    int search(vector<int>& nums, int target) {
        int r = nums.size()-1, l=0;
        int mid;
        while(r>=l){
            mid = l + (r-l)/2;
            if(nums[mid] == target) return mid;
            if(nums[mid] >= nums[l]){
                if(target<nums[mid] && target>=nums[l]){
                    r = mid-1;
                }else{
                    l = mid+1;
                }
            }else{
                if(target > nums[mid] && target<nums[l]){
                    l =mid+1;
                }else{
                    r = mid-1;
                }
            }
        }
        return -1;
    }
};
```

class Solution {
public:
    int search(vector<int>& nums, int target) {
        int r = nums.size()-1, l=0;
        int mid;
        while(r>=l){
            mid = l + (r-l)/2;

240.Search a 2D Matrix II



Accepted 130 / 130 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:01

Runtime 52 ms | Beats 63.24%

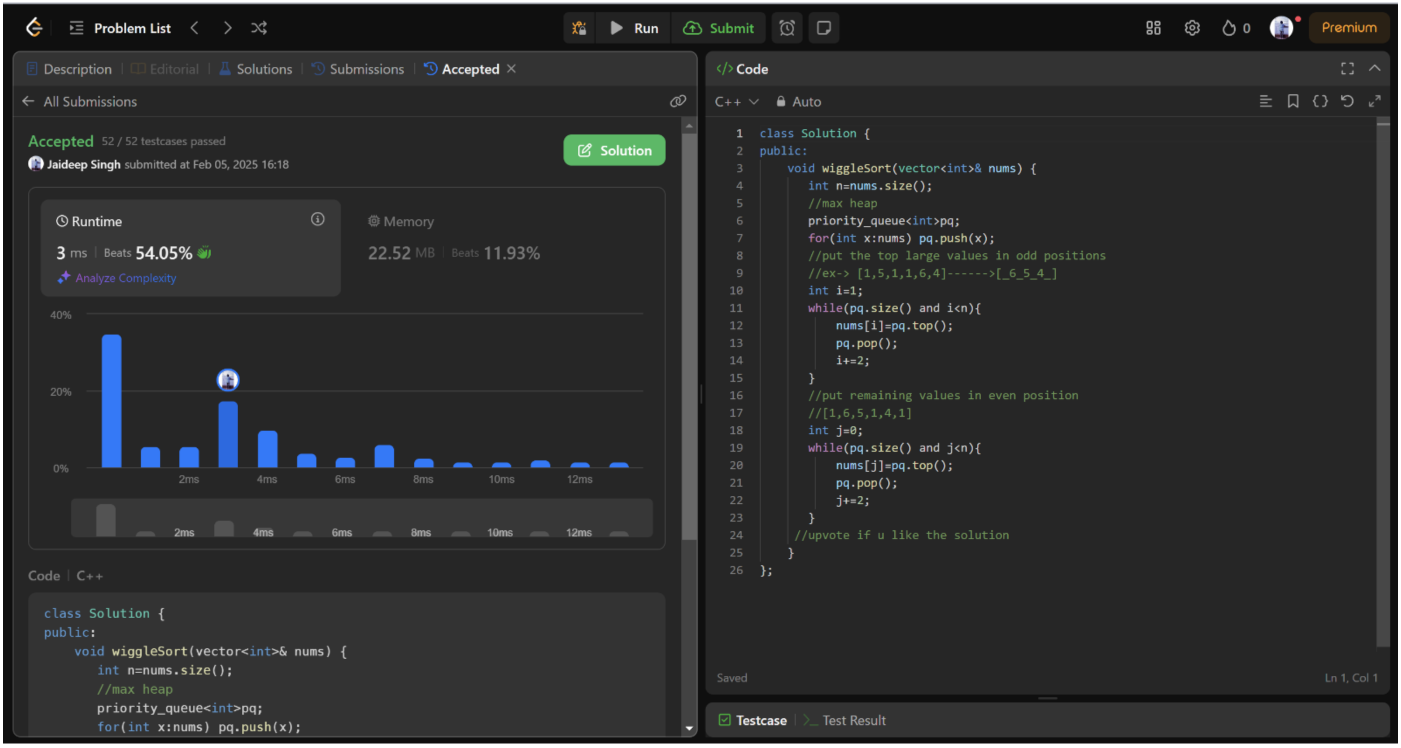
Memory 18.65 MB | Beats 67.47%

1.22% of solutions used 73 ms of runtime

```cpp
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int row = matrix.size(), col = matrix[0].size(), i = 0, j = col - 1;
        while(i < row && j >= 0){
            if(matrix[i][j] == target) return true;
            else if(matrix[i][j] > target) j--;
            else i++;
        }
        return false;
    }
};
```

```cpp
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int row = matrix.size(), col = matrix[0].size(), i = 0, j = col - 1;
        while(i < row && j >= 0){
            if(matrix[i][j] == target) return true;
            else if(matrix[i][j] > target) j--;
```

324.Wiggle Sort II



Accepted 52 / 52 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:18

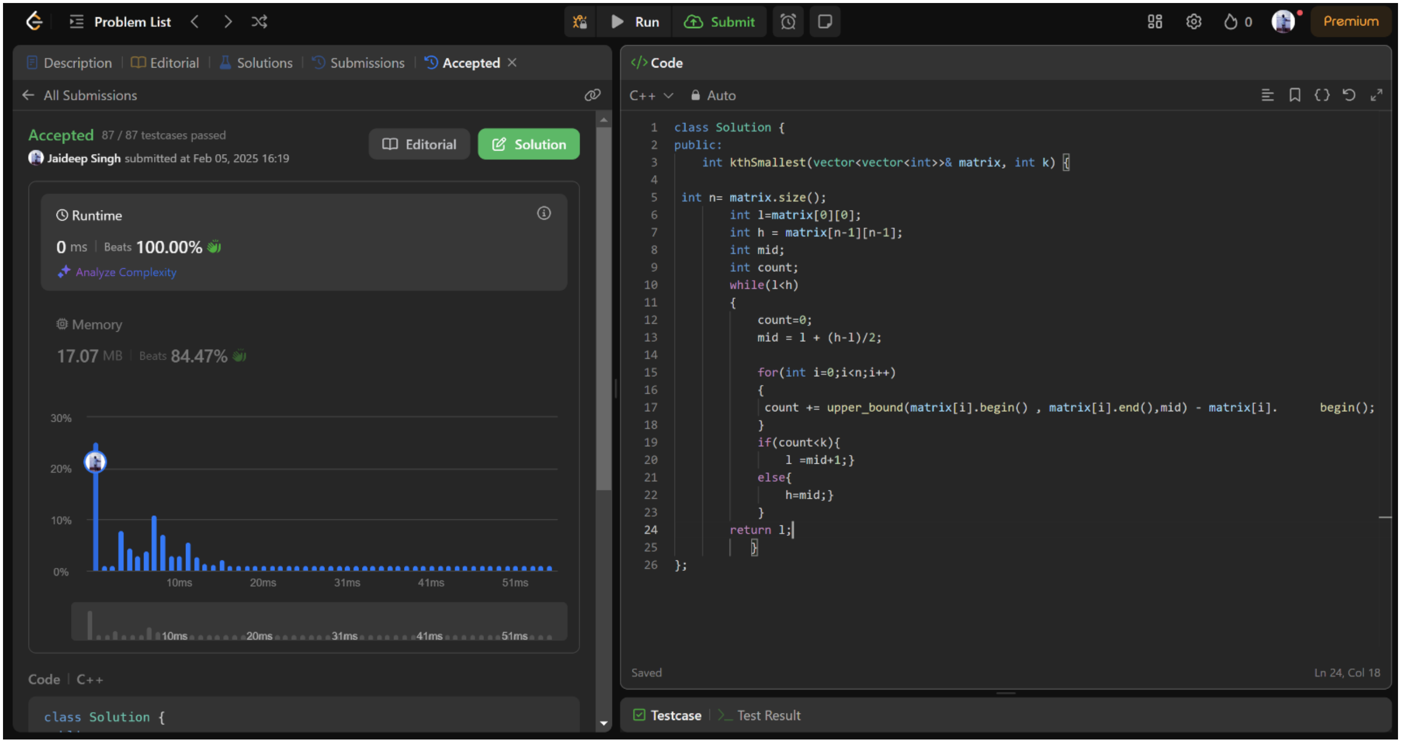
Runtime 3 ms | Beats 54.05%

Memory 22.52 MB | Beats 11.93%

```cpp
class Solution {
public:
    void wiggleSort(vector<int>& nums) {
       int n=nums.size();
       //max heap
       priority_queue<int>pq;
       for(int x:nums) pq.push(x);
       //put the top large values in odd positions
       //ex-> [1,5,1,1,6,4]------>[_6_5_4_]
       int i=1;
       while(pq.size() and i<n){
           nums[i]=pq.top();
           pq.pop();
           i+=2;
       }
       //put remaining values in even position
       //[1,6,5,1,4,1]
       int j=0;
       while(pq.size() and j<n){
           nums[j]=pq.top();
           pq.pop();
           j+=2;
       }
     //upvote if u like the solution
    }
};
```

```cpp
class Solution {
public:
    void wiggleSort(vector<int>& nums) {
       int n=nums.size();
       //max heap
       priority_queue<int>pq;
       for(int x:nums) pq.push(x);
```

378.Kth Smallest Element in a Sorted Matrix



Accepted 87 / 87 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:19

Runtime

0 ms | Beats 100.00%

Memory

17.07 MB | Beats 84.47%

Code | C++

```
class Solution {
```

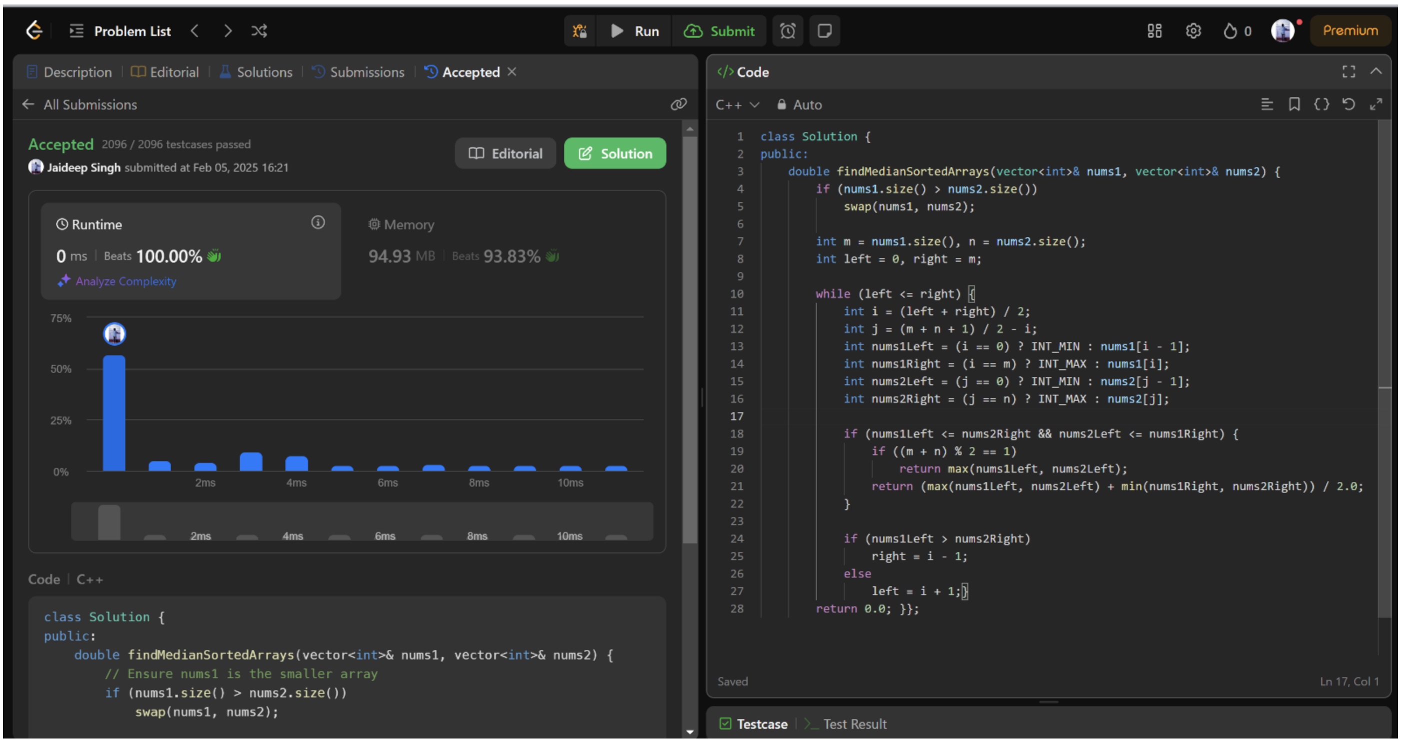
Code

C++ Auto

```
class Solution {
public:
    int kthSmallest(vector<vector<int>>& matrix, int k) {

 int n= matrix.size();
        int l=matrix[0][0];
        int h = matrix[n-1][n-1];
        int mid;
        int count;
        while(l<h)
        {
            count=0;
            mid = l + (h-l)/2;

            for(int i=0;i<n;i++)
            {
             count += upper_bound(matrix[i].begin() , matrix[i].end(),mid) - matrix[i].      begin();
            }
            if(count<k){
                l =mid+1;}
            else{
                h=mid;}
            }
        return l;
            }
};
```

Saved Ln 24, Col 18

4.Median of Two Sorted Arrays



Accepted 2096 / 2096 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:21

Runtime

0 ms | Beats 100.00%

Memory

94.93 MB | Beats 93.83%

Code | C++

```
class Solution {
public:
    double findMedianSortedArrays(vector<int>& nums1, vector<int>& nums2) {
        // Ensure nums1 is the smaller array
        if (nums1.size() > nums2.size())
            swap(nums1, nums2);
```

Code

C++ Auto

```
class Solution {
public:
    double findMedianSortedArrays(vector<int>& nums1, vector<int>& nums2) {
        if (nums1.size() > nums2.size())
            swap(nums1, nums2);

        int m = nums1.size(), n = nums2.size();
        int left = 0, right = m;

        while (left <= right) {
            int i = (left + right) / 2;
            int j = (m + n + 1) / 2 - i;
            int nums1Left = (i == 0) ? INT_MIN : nums1[i - 1];
            int nums1Right = (i == m) ? INT_MAX : nums1[i];
            int nums2Left = (j == 0) ? INT_MIN : nums2[j - 1];
            int nums2Right = (j == n) ? INT_MAX : nums2[j];

            if (nums1Left <= nums2Right && nums2Left <= nums1Right) {
                if ((m + n) % 2 == 1)
                    return max(nums1Left, nums2Left);
                return (max(nums1Left, nums2Left) + min(nums1Right, nums2Right)) / 2.0;
            }

            if (nums1Left > nums2Right)
                right = i - 1;
            else
                left = i + 1;}
        return 0.0; }};
```

Saved Ln 17, Col 1