

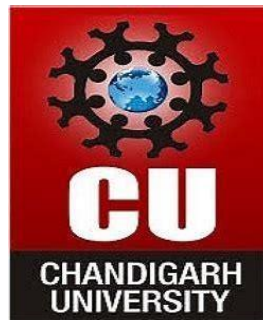


**CHANDIGARH
UNIVERSITY**
Discover. Learn. Empower.

**NAAC
GRADE A+**
Accredited University

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

Problem List < > 🔍

Run Submit ⌛

0 0 0 Premium

Description | Editorial | Solutions | Submissions | Accepted ✕

< All Submissions 🔗

Accepted 73 / 73 testcases passed
Jaideep Singh submitted at Feb 05, 2025 15:37 [Solution](#)

Runtime ⌚
0 ms | Beats 100.00% 🏆
[Analyze Complexity](#)

Memory 📦
8.27 MB | Beats 96.75% 🏆

Runtime (ms)	Percentage (%)
2ms	~35%

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;
            int largeMask = 0;
            char ch = s[i];
            int chint = 0;
            if(ch >= 65 && ch <= 90){
                chint = ch - 'A';
            }
        }
    }
};
```

Code

C++ Auto

Ln 1, Col 1

Testcase > Test Result

Case 1 Case 2 Case 3 +

s =

"YazaAay"

</> Source ⓘ

190. Reverse Bits

Description | Editorial | Accepted ✕ | Solutions | Submissions

< > 🔍

Run Submit ⌛

0 0 0 Premium

Description | Editorial | Accepted ✕ | Solutions | Submissions

< All Submissions 🔗

Accepted 600 / 600 testcases passed
Jaideep Singh submitted at Feb 05, 2025 15:40 [Editorial](#) [Solution](#)

Runtime ⌚
4 ms | Beats 33.11%
[Analyze Complexity](#)

Memory 📦
7.88 MB | Beats 29.58%

Runtime (ms)	Percentage (%)
4ms	~35%

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;
            n >>= 1;
        }
        return ans;
    }
};
```

Code

C++ Auto

Ln 1, Col 1

Testcase > Test Result

Case 1 Case 2 +

n =

00000010100101000001111010011100

</> Source ⓘ

191. Number of 1 Bits

Problem List < > 🔍

Run Submit ⌛

0 0 Premium

Description Editorial Solutions Accepted Submissions

All Submissions

Accepted 598 / 598 testcases passed

Jaideep Singh submitted at Feb 05, 2025 15:41

Editorial Solution

Runtime 0 ms | Beats 100.00%

Memory 8.27 MB | Beats 47.62%

Analyze Complexity

Runtime (ms)	Beats (%)
0	100.00%
1	~0%
2	~0%
3	~0%
4	~0%

Code | C++

```
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++;
        }
    }
};
```

Code

```
1 class Solution {
2 public:
3     int hammingWeight(int n) {
4         int c=0,i=0,m=n;
5         long bit;
6         while(m>0)
7         {
8             bit=pow(2,i);
9             if((n&bit)>=1)
10                 c++;
11             m/=2;
12             i++;
13         }
14     }
15 }
```

Testcase Test Result

Case 1 Case 2 Case 3 +

n =

11

</> Source

53. Maximum Subarray

Problem List < > 🔍

Run Submit ⌛

0 0 Premium

Description Editorial Solutions Accepted Submissions

All Submissions

Accepted 210 / 210 testcases passed

Jaideep Singh submitted at Sep 04, 2024 18:41

Editorial Solution

Runtime 85 ms | Beats 5.49%

Memory 70.40 MB | Beats 99.93%

Analyze Complexity

Runtime (ms)	Beats (%)
85	5.49%
1	~0%
2	~0%
3	~0%
4	~0%
5	~0%

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++)
```

Code

```
1 class Solution {
2 public:
3     int maxSubArray(vector<int>& nums) {
4         int l=0, r=0;
5         int res = -99999;
6         int sum = 0;
7         while (r < nums.size()) {
8             while (sum < 0) {
9                 sum -= nums[l];
10                l++;
11            }
12            sum += nums[r];
13        }
14    }
15 }
```

Testcase Test Result

Case 1 Case 2 Case 3 +

nums =

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

</> Source

240. Search a 2D Matrix II

Problem List < > ✕

Run Submit

Description Editorial Solutions Accepted ✕ Submissions

All Submissions

Accepted 130 / 130 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:01

Editorial Solution

Runtime 52 ms | Beats 63.24% @ Memory 18.65 MB | Beats 67.47%

Analyze Complexity

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--;
            else i++;
        }
        return false;
    }
};
```

Saved Ln 1, Col 1

Testcase Test Result

372. Super Pow

Problem List < > ✕

Run Submit

Description Editorial Solutions Accepted ✕ Submissions

All Submissions

Accepted 57 / 57 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:03

Solution

Runtime 5 ms | Beats 11.42% @ Memory 15.27 MB | Beats 52.19%

Analyze Complexity

Code | C++

```
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;
    }
};
```

Saved Ln 1, Col 1

932. Beautiful Array

DescriptionEditorialSolutionsAccepted XSubmissions

All Submissions

Accepted 38 / 38 testcases passed
Jaideep Singh submitted at Feb 05, 2025 16:04

EditorialSolution

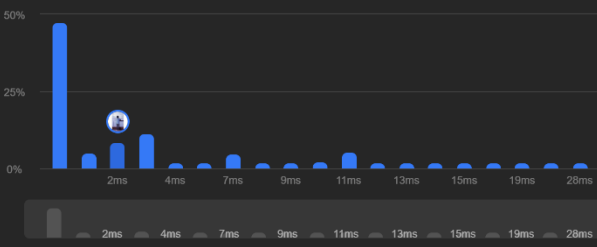
Runtime

2 ms | Beats 47.48%

Analyze Complexity

Memory

9.95 MB | Beats 74.37%



Code | C++

```
class Solution {
public:
    vector<int> beautifulArray(int n)
    {
        vector<int>ans;
        ans.push_back(1);
        while(ans.size()!=n)

```

Code

C++Auto

```
1 class Solution {
2 public:
3     vector<int> beautifulArray(int n)
4     {
5         vector<int>ans;
6         ans.push_back(1);
7         while(ans.size()!=n)
8         {
9             vector<int>temp;
10            for (auto it: ans)
11            {
12                if (it*2-1<=n)
13                    temp.push_back(it*2-1);
14            }
15            for (auto it: ans)
16            {
17                if(it*2<=n)
18                    temp.push_back(it*2);
19            }
20            ans=temp;
21        }
22        return ans;
23    }
24};
```

SavedLn 1, Col 1

TestcaseTest Result

218. The Skyline Problem

DescriptionEditorialSolutionsSubmissionsAccepted X

All Submissions

Accepted 44 / 44 testcases passed
Jaideep Singh submitted at Feb 05, 2025 16:05

EditorialSolution

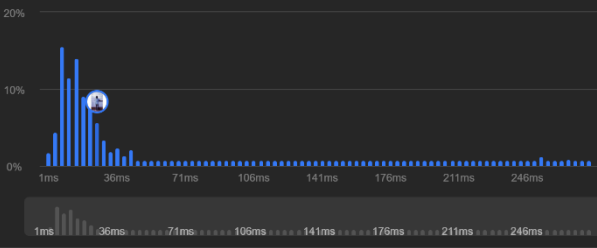
Runtime

28 ms | Beats 31.02%

Analyze Complexity

Memory

30.95 MB | Beats 32.90%



Code | C++

```
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>>>

```

Code

C++Auto

```
1 class Solution {
2 public:
3     vector<vector<int>> getSkyline(vector<vector<int>>& v) {
4         std::ios_base::sync_with_stdio(false);
5         std::cout.tie(nullptr);
6         std::cin.tie(nullptr);
7         int n=v.size();
8         int end=v[n-1][1];
9         //v.push_back({end+1,end+1,0});
10        unordered_map<int,int> m;
11        map<int,int> ans;
12        priority_queue<pair<int,int>, vector<pair<int,int>>, greater<pair<int,int>>>
13        pq;
14        set<int> s;
15        s.insert(0);
16        for(int i=0;i<=n;i++)
17        {
18            while(!pq.empty()&&(i==n||pq.top().first<v[i][0]))
19            {
20                int e=pq.top().first;
21                int h=pq.top().second;
22                pq.pop();
23                if(m[h]==e)
24                    {cout<<h<<" ";
25                     s.erase(h);
26                }
27                if(*s.rbegin()<h) ans[e]=*s.rbegin();
28                //cout<<*s.rbegin()<<" ";
29                if(i<n){
30                    if(*s.rbegin()<v[i][2])

```

SavedLn 1, Col 1

TestcaseTest Result

493. Reverse Pairs

Problem List < > ✕

Run Submit ⌵

Premium

Description Editorial Solutions Submissions Accepted ✕

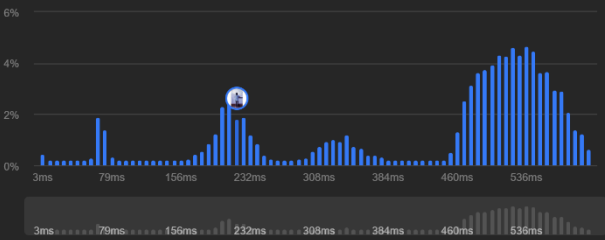
All Submissions

Accepted 140 / 140 testcases passed
Jaideep Singh submitted at Feb 05, 2025 16:07

Editorial Solution

Runtime 218 ms | Beats 85.14% 🏆
Analyze Complexity

Memory 112.94 MB | Beats 82.58% 🏆



3ms 79ms 156ms 232ms 308ms 384ms 460ms 536ms

Code | C++

```
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]){
                temp[k++] = nums[i++];
            }
            else {
                temp[k++] = nums[j++];
            }
        }
        while(i<=mid) temp[k++] = nums[i++];
        while(j<=right) temp[k++] = nums[j++];
        for(int i=left; i<=right; i++) nums[i] = temp[i-left];
    }
};
```

Code

C++ Auto

```
1 class Solution {
2 public:
3     int reversePairs(vector<int>& nums) {
4         int ans = 0, n = nums.size();
5         sect(0, n-1, nums, ans);
6         return ans;
7     }
8     void sect(int i, int j, vector<int>& a, int &ans){
9         if(i>=j) return;
10        int m = i+(j-i)/2;
11        sect(i, m, a, ans);
12        sect(m+1, j, a, ans);
13        countPairs(i, m, j, a, ans);
14    }
15    void countPairs(int i, int m, int j, vector<int>& a, int &ans){
16        int l = i, r = m+1;
17        while(l<=m && r<=j){
18            if(a[l]>2LL*a[r]){
19                ans += (m-l+1);
20                r++;
21            }
22            else l++;
23        }
24        merge(i, m, j, a);
25    }
26    void merge(int left, int mid, int right, vector<int>& nums){
27        vector<int> temp(right - left + 1);
28        int i = left, j=mid+1, k=0;
29        while(i<=mid && j<=right){
30            if(nums[i]<=nums[j]){
```

Saved Ln 1, Col 1

2407. Longest Increasing Subsequence II

Problem List < > ✕

Run Submit ⌵

Premium

Description Editorial Solutions Submissions Accepted ✕

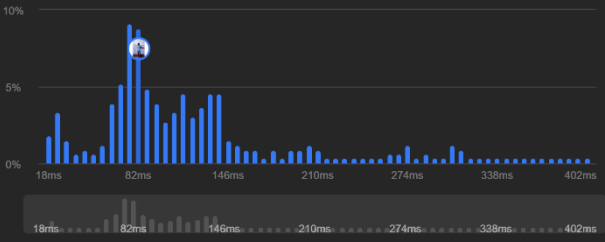
All Submissions

Accepted 84 / 84 testcases passed
Jaideep Singh submitted at Feb 05, 2025 16:08

Solution

Runtime 82 ms | Beats 68.39% 🏆
Analyze Complexity

Memory 63.54 MB | Beats 58.05% 🏆



18ms 82ms 146ms 210ms 274ms 338ms 402ms

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;
        }
        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;
        }
        ll mid=(low + high)/2;
        if(i<=mid) update(seg, low, mid, i, val, 2*ind + 1);
        else update(seg, mid+1, high, i, val, 2*ind + 2);
    }
};
```

Code

C++ Auto

```
1 #define ll int
2 class Solution {
3 public:
4     void build(vector<ll>& seg, vector<ll>& a, ll low, ll high, ll ind){
5         if(low==high){
6             seg[ind]=a[low];
7             return;
8         }
9         ll mid=(low + high)/2;
10        build(seg, a, low, mid, 2*ind + 1);
11        build(seg, a, mid+1, high, 2*ind + 2);
12        seg[ind]=max(seg[2*ind + 1], seg[2*ind + 2]);
13    }
14    ll query(vector<ll>& seg, ll low, ll high, ll x, ll y, ll ind){
15        if(x>high || y<low){
16            return INT_MIN;
17        }
18        if(low==x && high==y){
19            return seg[ind];
20        }
21        ll mid=(low + high)/2;
22        ll left=query(seg, low, mid, x, y, 2*ind + 1);
23        ll right=query(seg, mid+1, high, x, y, 2*ind + 2);
24        return max(left, right);
25    }
26    void update(vector<ll>& seg, ll low, ll high, ll i, ll val, ll ind){
27        if(low==high){
28            seg[ind]=max(seg[ind], val);
29            return;
30        }
31        ll mid=(low + high)/2;
32        if(i<=mid) update(seg, low, mid, i, val, 2*ind + 1);
33        else update(seg, mid+1, high, i, val, 2*ind + 2);
34    }
35 }
```

Saved Ln 1, Col 1

Testcase > Test Result

88. Merge Sorted Array

Problem List < > 🔍

Run Submit ⌂

Premium

Description Editorial Solutions Submissions Accepted ×

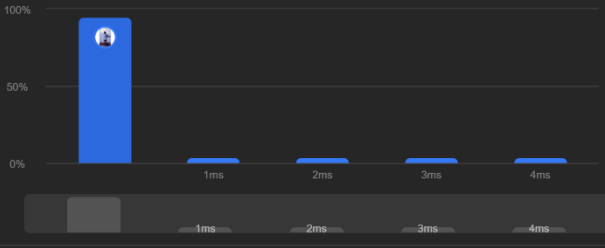
All Submissions

Accepted 59 / 59 testcases passed
Jaideep Singh submitted at Feb 05, 2025 16:09

Editorial Solution

Runtime 0 ms | Beats 100.00% 🏆
Analyze Complexity

Memory 12.28 MB | Beats 70.78% 🏆



1ms 2ms 3ms 4ms

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;
    }
};
```

C++ Auto

```
1 #include <vector>
2
3 class Solution {
4 public:
5     void merge(std::vector<int>& nums1, int m, std::vector<int>& nums2, int n) {
6         // Correct variable declarations
7         int p1 = m - 1;
8         int p2 = n - 1;
9         int p = m + n - 1;
10
11         // Merge from the back
12         while (p1 >= 0 && p2 >= 0) {
13             if (nums1[p1] > nums2[p2]) {
14                 nums1[p] = nums1[p1];
15                 p1--;
16             } else {
17                 nums1[p] = nums2[p2];
18                 p2--;
19             }
20             p--;
21         }
22
23         // Copy remaining elements of nums2 into nums1 if any
24         while (p2 >= 0) {
25             nums1[p] = nums2[p2];
26             p2--;
27             p--;
28         }
29     }
30 };
```

Saved Ln 1, Col 1

Testcase > Test Result

278. First Bad Version

Problem List < > 🔍

Run Submit ⌂

Premium

Description Editorial Solutions Submissions Accepted ×

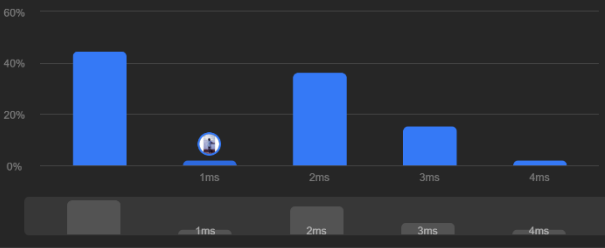
All Submissions

Accepted 24 / 24 testcases passed
Jaideep Singh submitted at Feb 05, 2025 16:11

Editorial Solution

Runtime 1 ms | Beats 55.13% 🏆
Analyze Complexity

Memory 7.90 MB | Beats 69.23% 🏆



1ms 2ms 3ms 4ms

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 ;
        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 ;
    }
};
```

C++ Auto

```
1 // The API isBadVersion is defined for you.
2 // bool isBadVersion(int version);
3
4 class Solution {
5 public:
6     int firstBadVersion(int n) {
7         long long int start = 0 ;
8         long long int end = n ;
9         long long int minBadVersion ;
10
11         while (start <= end)
12         {
13             long long int mid = (start + end) / 2 ;
14
15             if (isBadVersion(mid) == 1)
16             {
17                 minBadVersion = mid ;
18                 end = mid - 1 ;
19             }
20             else
21             {
22                 start = mid + 1 ;
23             }
24         }
25         return minBadVersion ;
26     }
27 };
```

Saved Ln 1, Col 1

Testcase > Test Result

75. Sort Colors

Problem List < > ✕ Run Submit 📄

Description Editorial Solutions Submissions Accepted ✕

All Submissions


Accepted 88 / 88 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:11

Editorial Solution

Runtime 0 ms | Beats 100.00% 🏆
Memory 11.78 MB | Beats 31.40%

Analyze Complexity



Code | C++

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

Saved Ln 1, Col 1

Testcase Test Result

347. Top K Frequent Elements

Problem List < > ✕ Run Submit 📄

Description Editorial Solutions Submissions Accepted ✕

All Submissions

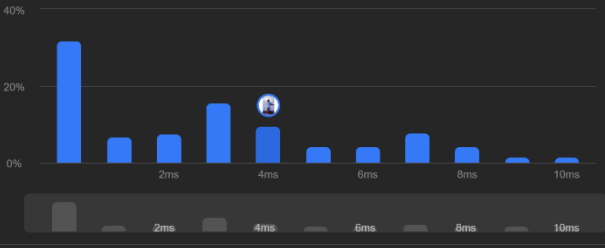
Accepted 21 / 21 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:12

Editorial Solution

Runtime 4 ms | Beats 37.84%
Memory 17.92 MB | Beats 48.42%

Analyze Complexity



Code | C++

```
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;
    }
};
```

Saved Ln 1, Col 1

Testcase Test Result

215. Kth Largest Element in an Array

Problem List < > ✕ Run Submit ⌚ 📄 Premium

Description Editorial Solutions Submissions Accepted ✕

← All Submissions

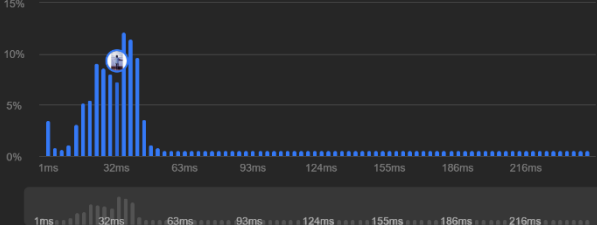
Accepted 42 / 42 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:13

Editorial Solution

Runtime 34 ms | Beats 49.29%
Analyze Complexity

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();
    }
};
```

Code

```
1 class Solution {
2 public:
3     int findKthLargest(vector<int>& nums, int k) {
4         priority_queue<int> heap(nums.begin(), nums.end());
5         while (--k) heap.pop();
6         return heap.top();
7     }
8 };
```

Saved Ln 1, Col 1

Testcase > Test Result

162. Find Peak Element

Problem List < > ✕ Run Submit ⌚ 📄 Premium

Description Editorial Solutions Submissions Accepted ✕

← All Submissions

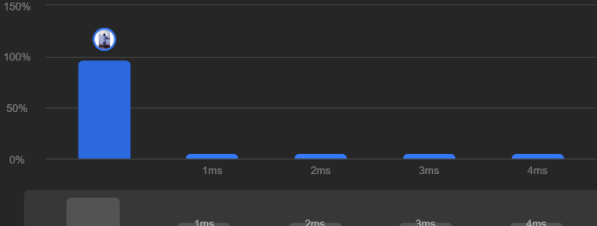
Accepted 68 / 68 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:14

Editorial Solution

Runtime 0 ms | Beats 100.00% 🏆
Analyze Complexity

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;
    }
}
```

Code

```
1 class Solution {
2     public int findPeakElement(int[] nums) {
3         int i = 0, j = nums.length - 1;
4         while(i < j){
5             int mid = (i + j)/2;
6             boolean PeakOnLeft = nums[mid] > nums[mid + 1];
7             if(PeakOnLeft){
8                 j = mid;
9             } else {
10                i = mid + 1;
11            }
12        }
13        return i;
14    }
15 }
```

Saved Ln 1, Col 1

Testcase > Test Result

56. Merge Intervals

Problem List < > ✕ Run Submit 📄 Premium

Description | Editorial | Solutions | Submissions | Accepted ✕

< All Submissions

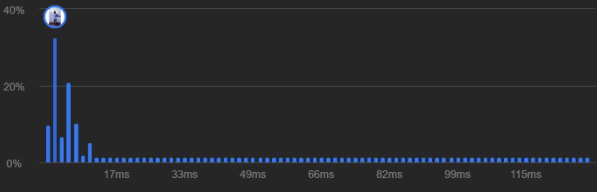
Accepted 171 / 171 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:16

Editorial Solution

Runtime 3 ms | Beats 90.04% 🏆
Analyze Complexity

Memory 23.95 MB | Beats 34.82%
Analyze Complexity



Code | C++

```
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++) {
```

```
<< Code
C++ v Auto
1 class Solution {
2 public:
3     vector<vector<int>> merge(vector<vector<int>>& a) {
4         sort(a.begin(), a.end());
5         vector<vector<int>> ans = {a[0]};
6
7         for (int i = 1; i < a.size(); i++) {
8             if (ans.back()[1] >= a[i][0]) {
9                 ans.back()[1] = max(ans.back()[1], a[i][1]);
10            } else {
11                ans.push_back(a[i]);
12            }
13        }
14        return ans;
15    }
16 };
```

Saved Ln 1, Col 1

Testcase Test Result

33. Search in Rotated Sorted Array

Problem List < > ✕ Run Submit 📄 Premium

Description | Editorial | Solutions | Submissions | Accepted ✕

< All Submissions

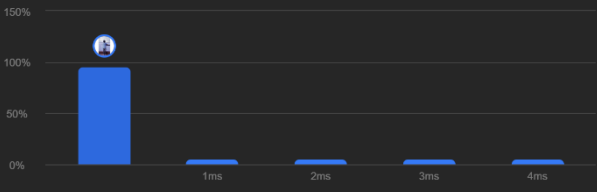
Accepted 196 / 196 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:21

Editorial Solution

Runtime 0 ms | Beats 100.00% 🏆
Analyze Complexity

Memory 15.18 MB | Beats 66.29% 🏆



Code | C++

```
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;
```

```
<< Code
C++ v Auto
1 class Solution {
2 public:
3     int search(vector<int>& nums, int target) {
4         int r = nums.size()-1, l=0;
5         int mid;
6         while(r>=l){
7             mid = l + (r-l)/2;
8             if(nums[mid] == target) return mid;
9             if(nums[mid] >= nums[l]){
10                 if(target<nums[mid] && target>=nums[l]){
11                     r = mid-1;
12                 }else{
13                     l = mid+1;
14                 }
15             }else{
16                 if(target > nums[mid] && target<nums[l]){
17                     l =mid+1;
18                 }else{
19                     r = mid-1;
20                 }
21             }
22         }
23         return -1;
24     }
25 };
```

Saved Ln 1, Col 1

Testcase Test Result

240. Search a 2D Matrix II

Problem List < > 🔍

Run Submit 📄

🔧 ⚙️ 🔔 0 🧑 Premium

Description Editorial Solutions Submissions Accepted ×

< All Submissions 🔗

Accepted 130 / 130 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:01

Editorial Solution

Runtime 52 ms | Beats 63.24% 🏆

Memory 18.65 MB | Beats 67.47% 🏆

Analyze Complexity

1.22% of solutions used 73 ms of runtime

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--;
            else i++;
        }
        return false;
    }
};
```

Code

C++ v Auto

```
1 class Solution {
2 public:
3     bool searchMatrix(vector<vector<int>>& matrix, int target) {
4         int row = matrix.size(), col = matrix[0].size(), i = 0, j = col - 1;
5         while(i < row && j >= 0){
6             if(matrix[i][j] == target) return true;
7             else if(matrix[i][j] > target) j--;
8             else i++;
9         }
10        return false;
11    }
12};
```

Saved Ln 1, Col 1

Testcase > Test Result

324. Wiggle Sort II

Problem List < > 🔍

Run Submit 📄

🔧 ⚙️ 🔔 0 🧑 Premium

Description Editorial Solutions Submissions Accepted ×

< All Submissions 🔗

Accepted 52 / 52 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:18

Solution

Runtime 3 ms | Beats 54.05% 🏆

Memory 22.52 MB | Beats 11.93%

Analyze Complexity

2ms 4ms 6ms 8ms 10ms 12ms

Code | C++

```
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
    }
};
```

Code

C++ v Auto

```
1 class Solution {
2 public:
3     void wiggleSort(vector<int>& nums) {
4         int n=nums.size();
5         //max heap
6         priority_queue<int>pq;
7         for(int x:nums) pq.push(x);
8         //put the top large values in odd positions
9         //ex-> [1,5,1,1,6,4]----->[_6_5_4_]
10        int i=1;
11        while(pq.size() and i<n){
12            nums[i]=pq.top();
13            pq.pop();
14            i+=2;
15        }
16        //put remaining values in even position
17        //[1,6,5,1,4,1]
18        int j=0;
19        while(pq.size() and j<n){
20            nums[j]=pq.top();
21            pq.pop();
22            j+=2;
23        }
24        //upvote if u like the solution
25    }
26};
```

Saved Ln 1, Col 1

Testcase > Test Result

378. Kth Smallest Element in a Sorted Matrix

Problem List < > ✕

Description Editorial Solutions Submissions Accepted ✕

All Submissions

Accepted 87 / 87 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:19

Editorial Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

17.07 MB | Beats 84.47%

30%

20%

10%

0%

0ms 10ms 20ms 31ms 41ms 51ms

Code | C++

class Solution {

Code

C++ v Auto

1 class Solution {

2 public:

3 int kthSmallest(vector<vector<int>>& matrix, int k) {

4

5 int n = matrix.size();

6 int l = matrix[0][0];

7 int h = matrix[n-1][n-1];

8 int mid;

9 int count;

10 while(l < h)

11 {

12 count = 0;

13 mid = l + (h - l) / 2;

14

15 for(int i = 0; i < n; i++)

16 {

17 count += upper_bound(matrix[i].begin(), matrix[i].end(), mid) - matrix[i].begin();

18 }

19 if(count < k){

20 l = mid + 1;

21 }

22 else{

23 h = mid;

24 }

25 return l;

26 };

Saved

Ln 24, Col 18

Testcase Test Result

4. Median of Two Sorted Arrays

Problem List < > ✕

Description Editorial Solutions Submissions Accepted ✕

All Submissions

Accepted 2096 / 2096 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:21

Editorial Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

94.93 MB | Beats 93.83%

75%

50%

25%

0%

0ms 2ms 4ms 6ms 8ms 10ms

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++ v Auto

1 class Solution {

2 public:

3 double findMedianSortedArrays(vector<int>& nums1, vector<int>& nums2) {

4 if (nums1.size() > nums2.size())

5 swap(nums1, nums2);

6

7 int m = nums1.size(), n = nums2.size();

8 int left = 0, right = m;

9

10 while (left <= right) {

11 int i = (left + right) / 2;

12 int j = (m + n + 1) / 2 - i;

13 int nums1Left = (i == 0) ? INT_MIN : nums1[i - 1];

14 int nums1Right = (i == m) ? INT_MAX : nums1[i];

15 int nums2Left = (j == 0) ? INT_MIN : nums2[j - 1];

16 int nums2Right = (j == n) ? INT_MAX : nums2[j];

17

18 if (nums1Left <= nums2Right && nums2Left <= nums1Right) {

19 if ((m + n) % 2 == 1)

20 return max(nums1Left, nums2Left);

21 return (max(nums1Left, nums2Left) + min(nums1Right, nums2Right)) / 2.0;

22 }

23

24 if (nums1Left > nums2Right)

25 right = i - 1;

26 else

27 left = i + 1;

28 return 0.0; }

Saved

Ln 17, Col 1

Testcase Test Result