NAAC GRADE A+
Accredited University

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF ENGINEERING

Department of Computer Science & Engineering

(BE-CSE-6th Sem)

CU
CHANDIGARH UNIVERSITY

Subject Name: Advanced Programming Lab - 2

Subject Code: 22CSP-351

Submitted to: Vishal Sir

Submitted by:

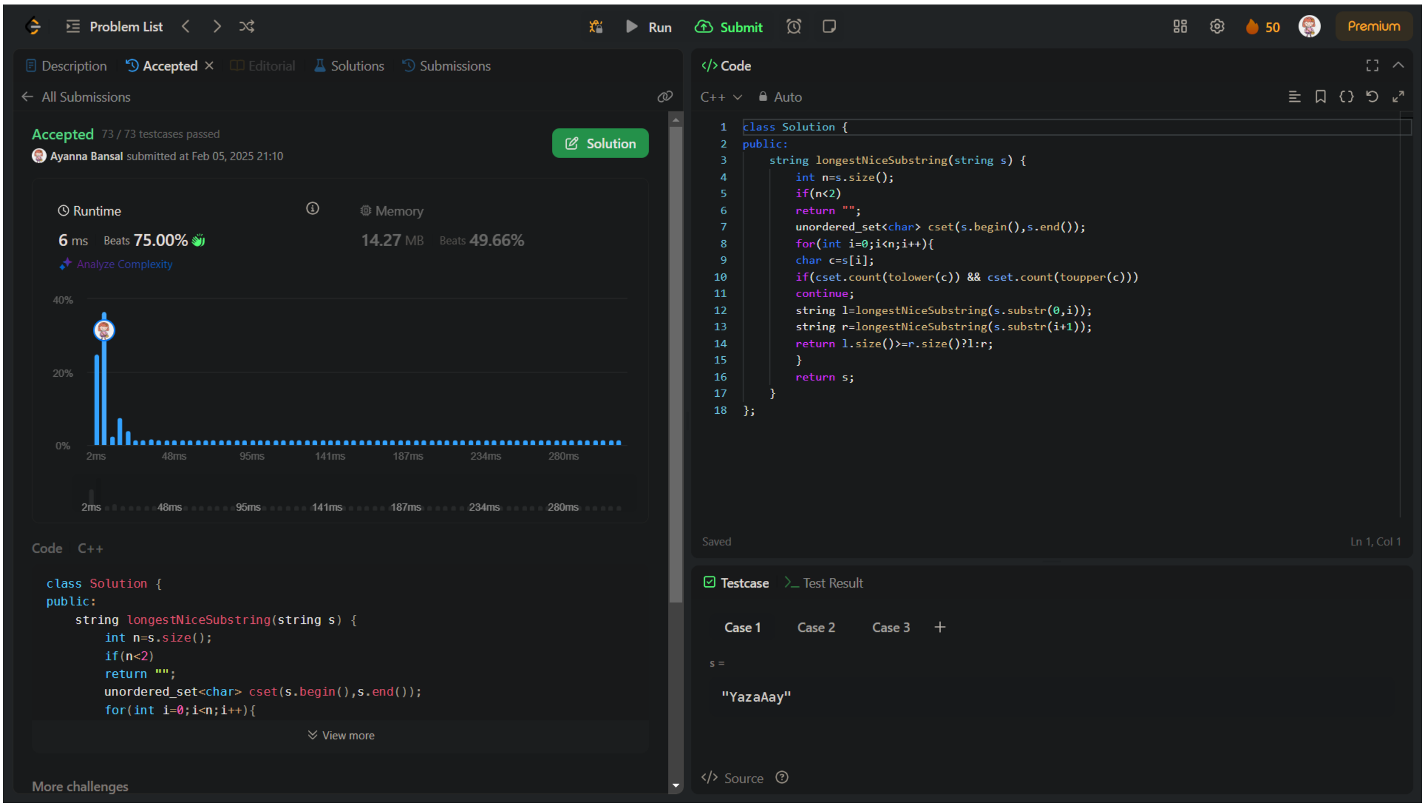
Name: Ayanna

UID:22BCS14805

Section: FL-IOT-604

Group: A

1763.Longest Nice Substring



Accepted 73 / 73 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:10

Runtime 6 ms Beats 75.00%

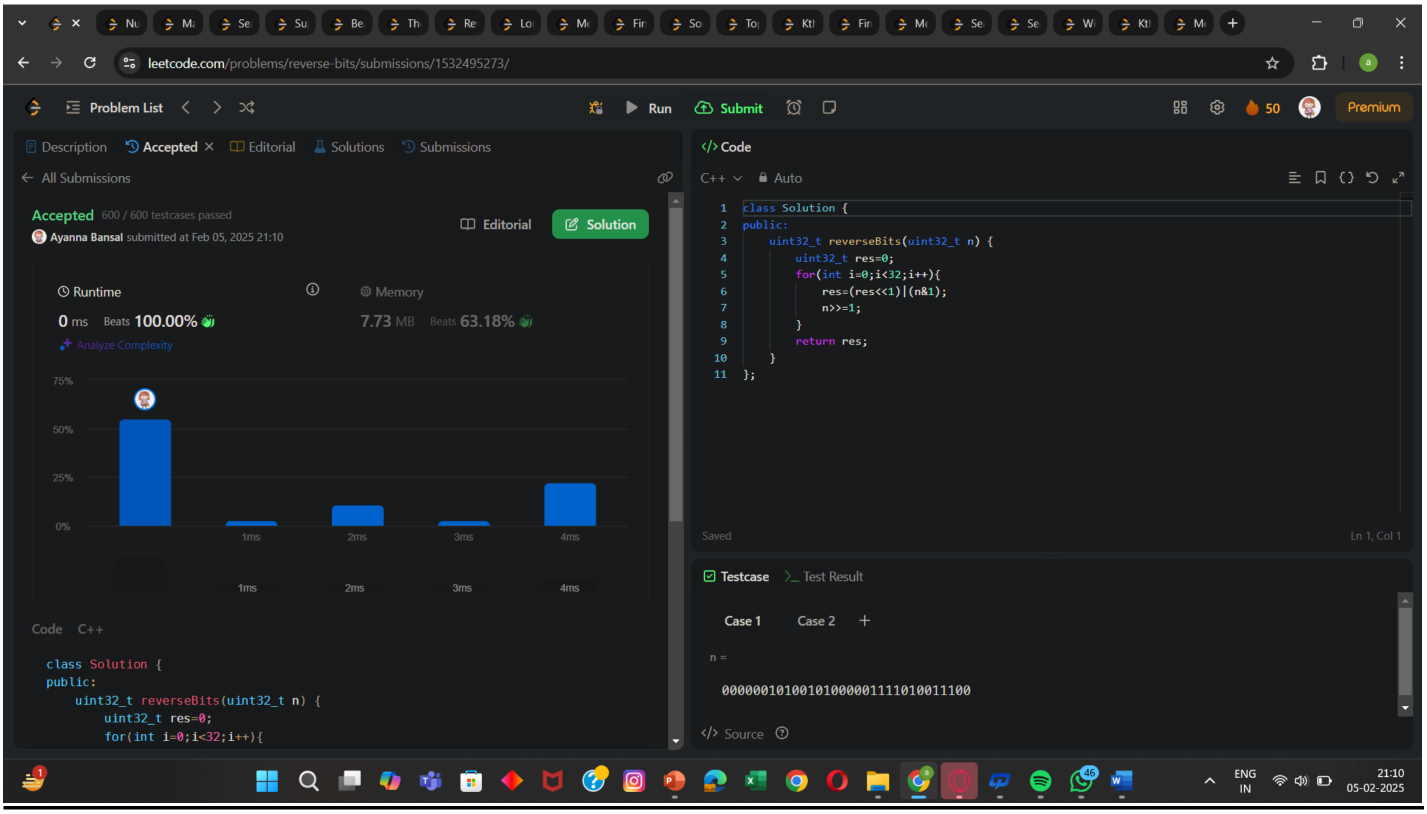
Memory 14.27 MB Beats 49.66%

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        int n=s.size();
        if(n<2)
        return "";
        unordered_set<char> cset(s.begin(),s.end());
        for(int i=0;i<n;i++){
        char c=s[i];
        if(cset.count(tolower(c)) && cset.count(toupper(c)))
        continue;
        string l=longestNiceSubstring(s.substr(0,i));
        string r=longestNiceSubstring(s.substr(i+1));
        return l.size()>=r.size()?l:r;
        }
        return s;
    }
};
```

s =

```
"YazaAay"
```

190.Reverse Bits



leetcode.com/problems/reverse-bits/submissions/1532495273/

Accepted 600 / 600 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:10

Runtime 0 ms Beats 100.00%

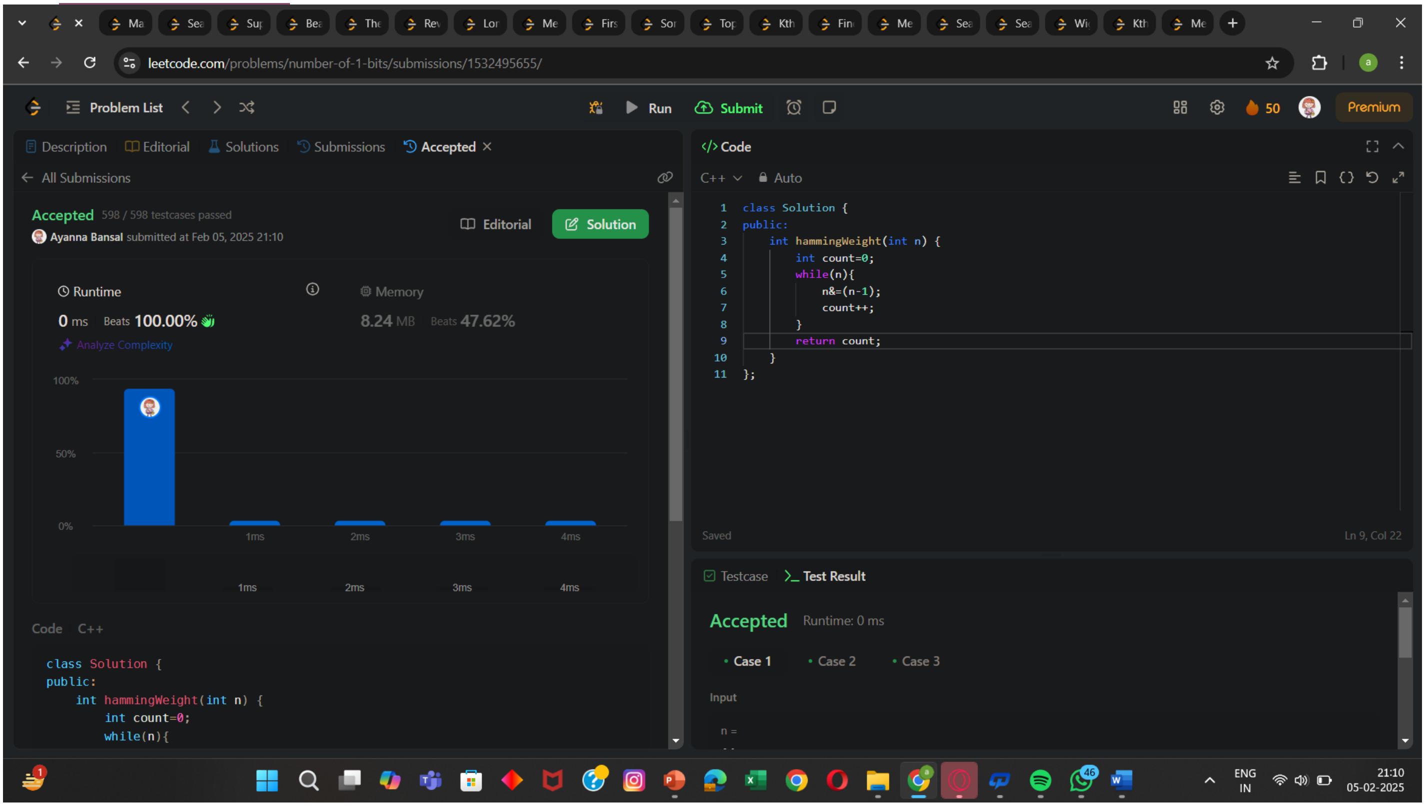
Memory 7.73 MB Beats 63.18%

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

n =

```
00000010100101000001111010011100
```

191.Number of 1 Bits



leetcode.com/problems/number-of-1-bits/submissions/1532495655/

Accepted 598 / 598 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:10

Runtime 0 ms Beats 100.00%

Memory 8.24 MB Beats 47.62%

```
class Solution {
public:
    int hammingWeight(int n) {
        int count=0;
        while(n){
            n&=(n-1);
            count++;
        }
        return count;
    }
};
```

Saved Ln 9, Col 22

Accepted Runtime: 0 ms

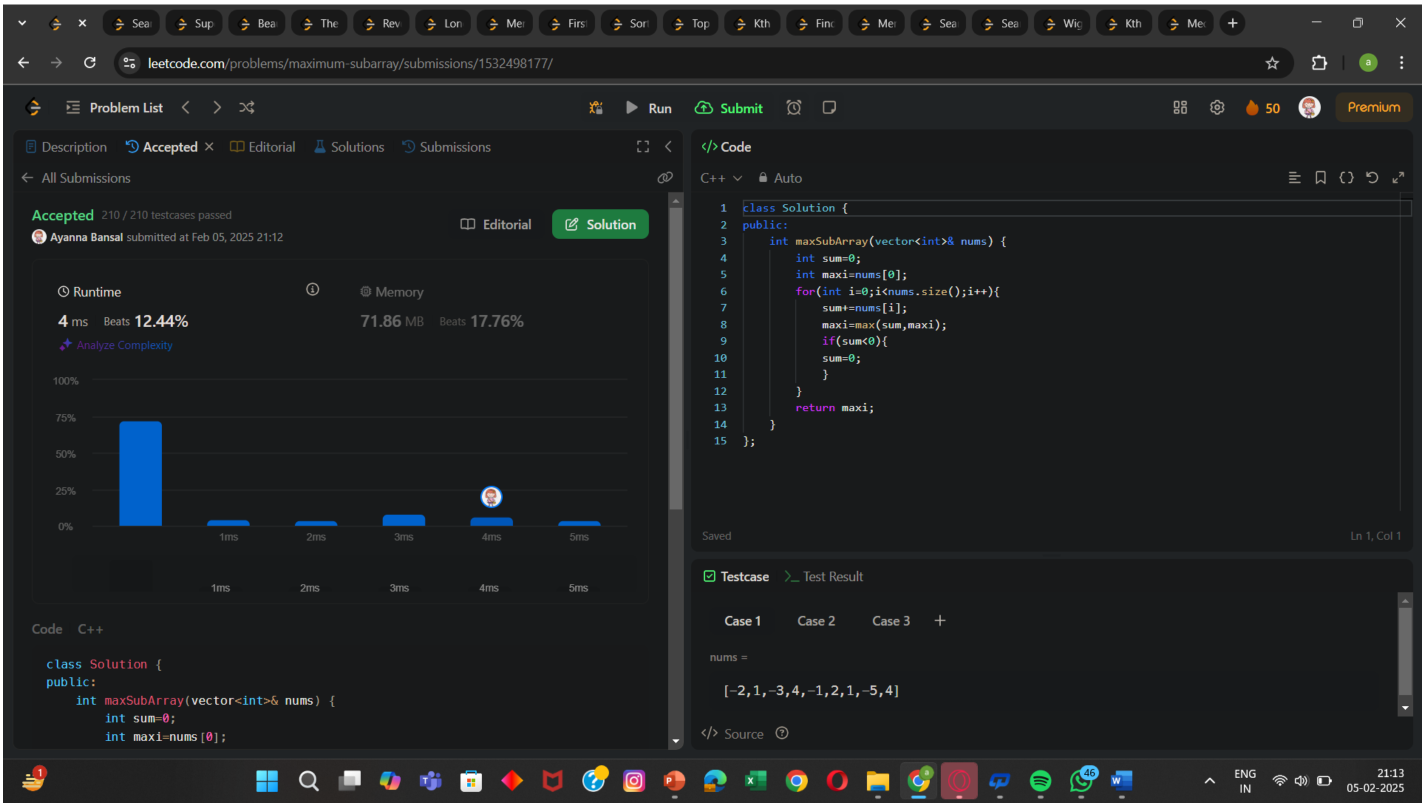
• Case 1 • Case 2 • Case 3

Input

n =

21:10
05-02-2025

53.Maximum Subarray



leetcode.com/problems/maximum-subarray/submissions/1532498177/

Accepted 210 / 210 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:12

Runtime 4 ms Beats 12.44%

Memory 71.86 MB Beats 17.76%

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int sum=0;
        int maxi=nums[0];
        for(int i=0;i<nums.size();i++){
            sum+=nums[i];
            maxi=max(sum,maxi);
            if(sum<0){
            sum=0;
            }
        }
        return maxi;
    }
};
```

Saved Ln 1, Col 1

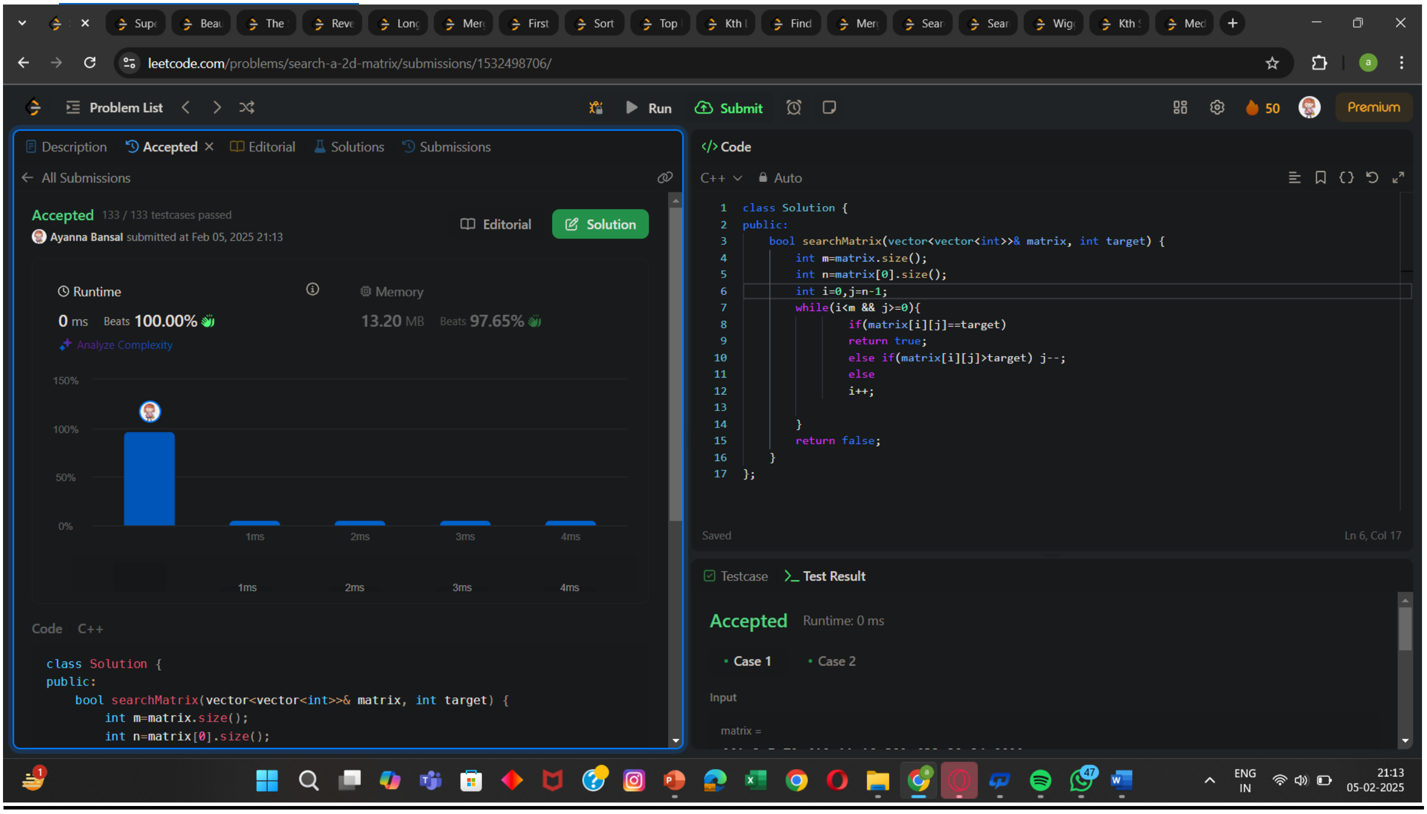
Case 1 Case 2 Case 3 +

nums =

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

21:13
05-02-2025

240.Search a 2D Matrix II



leetcode.com/problems/search-a-2d-matrix/submissions/1532498706/

Accepted 133 / 133 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:13

Runtime 0 ms Beats 100.00%

Memory 13.20 MB Beats 97.65%

Code C++

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

        }
        return false;
    }
};
```

Saved   Ln 6, Col 17

Test Result

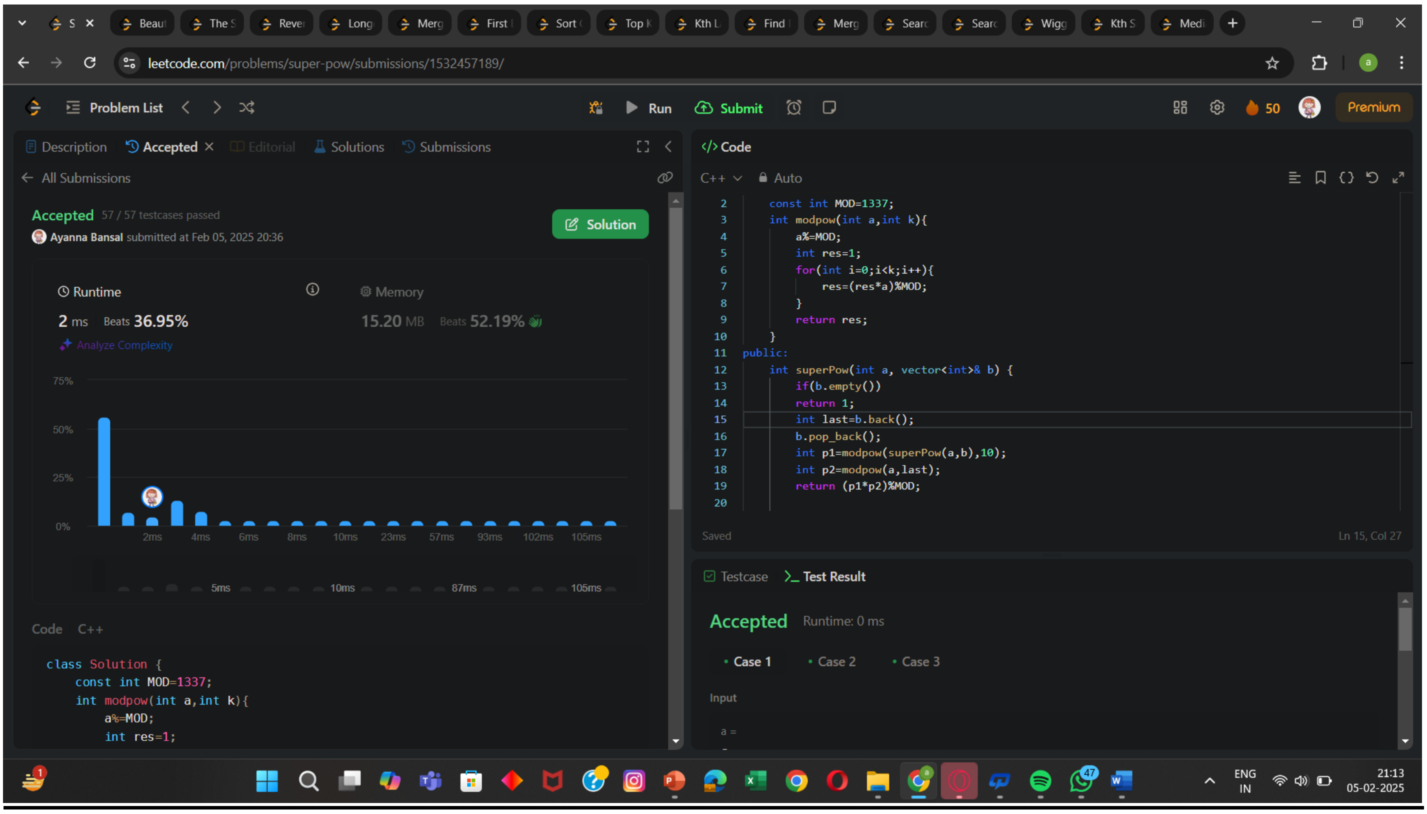
Accepted Runtime: 0 ms

Case 1 Case 2

Input

matrix =

372.Super Pow



leetcode.com/problems/super-pow/submissions/1532457189/

Accepted 57 / 57 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 20:36

Runtime 2 ms Beats 36.95%

Memory 15.20 MB Beats 52.19%

Code C++

```
class Solution {
    const int MOD=1337;
    int modpow(int a,int k){
        a%=MOD;
        int res=1;
        for(int i=0;i<k;i++){
            res=(res*a)%MOD;
        }
        return res;
    }
public:
    int superPow(int a, vector<int>& b) {
        if(b.empty())
        return 1;
        int last=b.back();
        b.pop_back();
        int p1=modpow(superPow(a,b),10);
        int p2=modpow(a,last);
        return (p1*p2)%MOD;

```

Saved   Ln 15, Col 27

Test Result

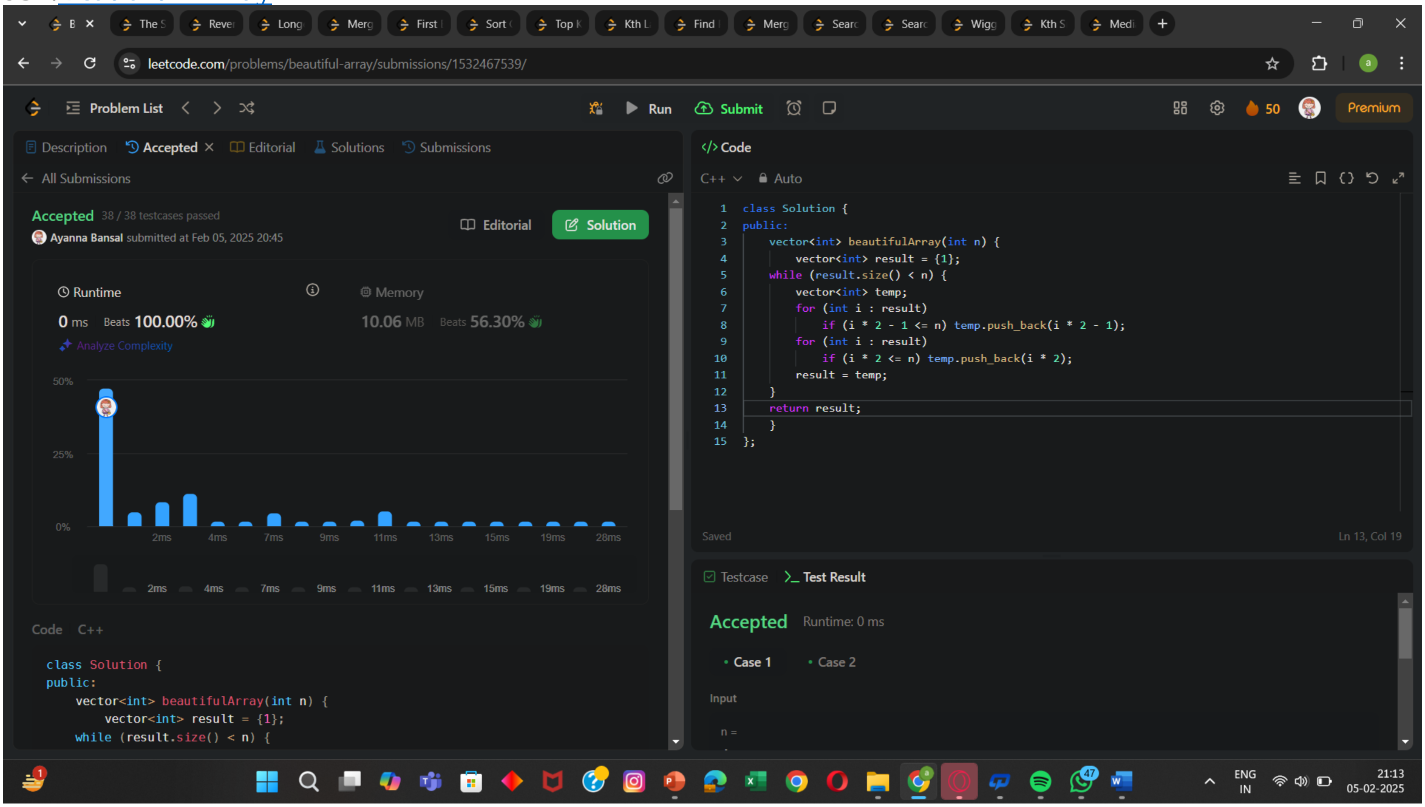
Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

a =

932.Beautiful Array



leetcode.com/problems/beautiful-array/submissions/1532467539/

Accepted 38 / 38 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 20:45

Runtime 0 ms Beats 100.00%

Memory 10.06 MB Beats 56.30%

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

Code C++

```cpp
class Solution {
public:
    vector<int> beautifulArray(int n) {
        vector<int> result = {1};
    while (result.size() < n) {
```

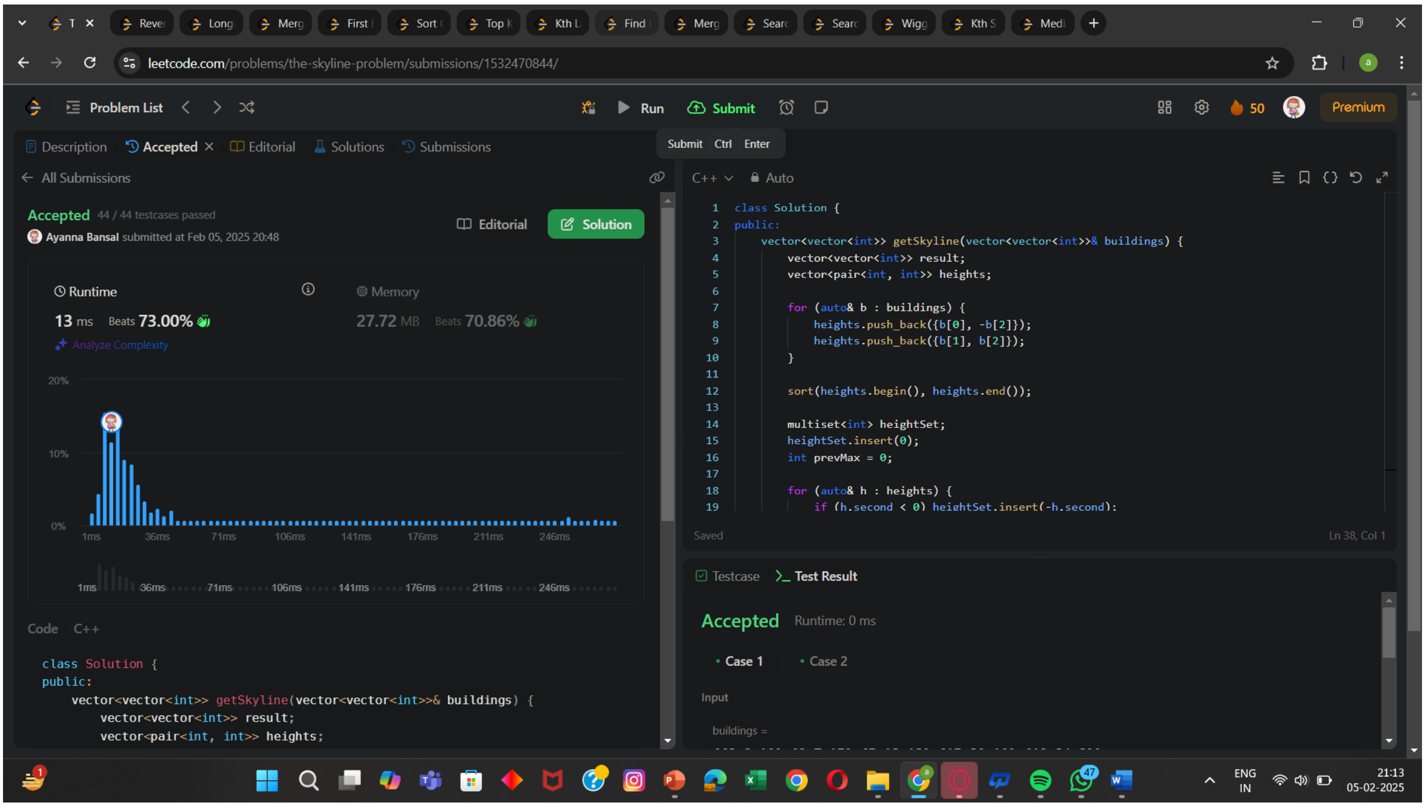
Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

n =

218.The Skyline Problem



leetcode.com/problems/the-skyline-problem/submissions/1532470844/

Accepted 44 / 44 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 20:48

Runtime 13 ms Beats 73.00%

Memory 27.72 MB Beats 70.86%

```cpp
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<vector<int>> result;
        vector<pair<int, int>> heights;

        for (auto& b : buildings) {
            heights.push_back({b[0], -b[2]});
            heights.push_back({b[1], b[2]});
        }

        sort(heights.begin(), heights.end());

        multiset<int> heightSet;
        heightSet.insert(0);
        int prevMax = 0;

        for (auto& h : heights) {
            if (h.second < 0) heightSet.insert(-h.second);
```

Code C++

```cpp
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<vector<int>> result;
        vector<pair<int, int>> heights;
```

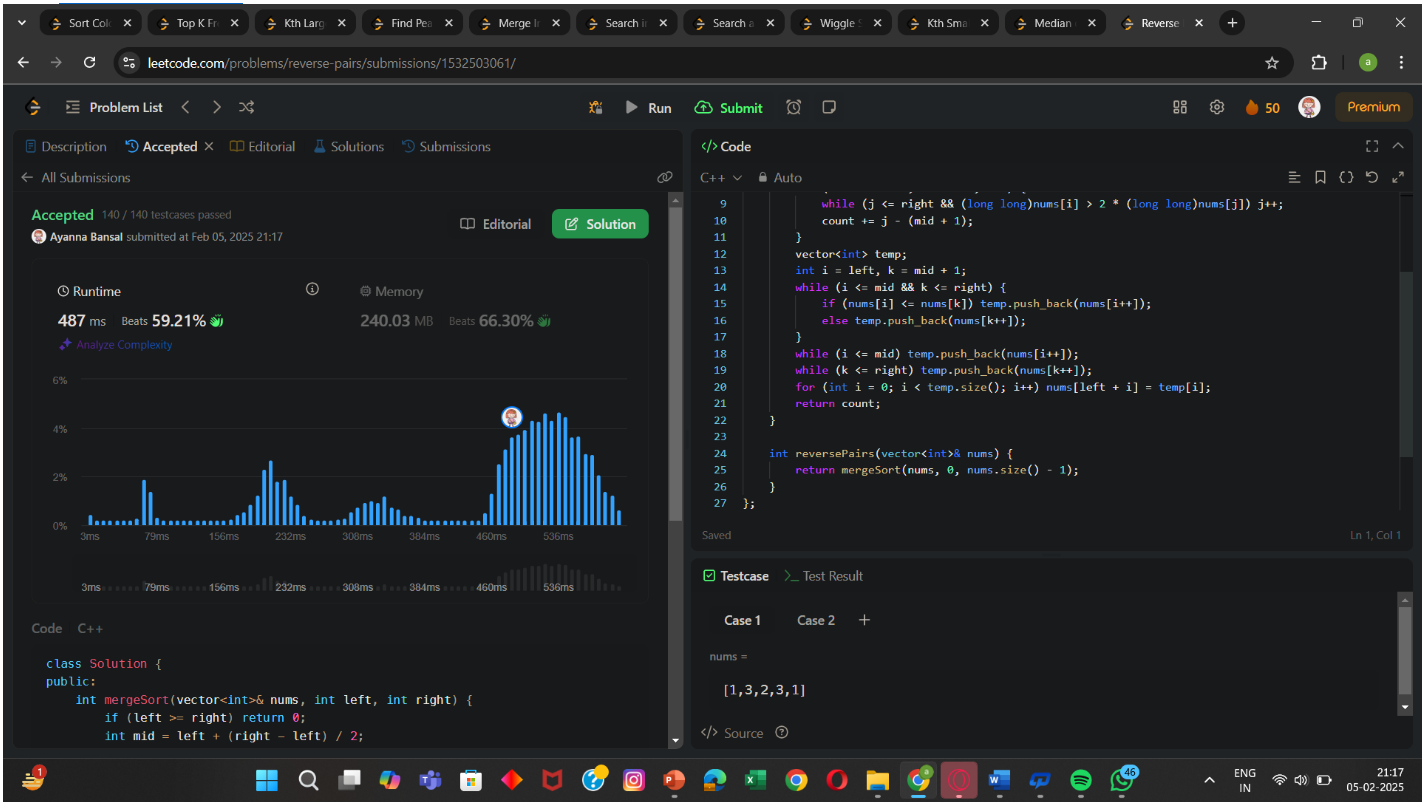
Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

buildings =

493.Reverse Pairs



Accepted 140 / 140 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:17

Runtime 487 ms Beats 59.21%

Memory 240.03 MB Beats 66.30%

Code C++

```
class Solution {
public:
    int mergeSort(vector<int>& nums, int left, int right) {
        if (left >= right) return 0;
        int mid = left + (right - left) / 2;
```

```
            while (j <= right && (long long)nums[i] > 2 * (long long)nums[j]) j++;
            count += j - (mid + 1);
        }
        vector<int> temp;
        int i = left, k = mid + 1;
        while (i <= mid && k <= right) {
            if (nums[i] <= nums[k]) temp.push_back(nums[i++]);
            else temp.push_back(nums[k++]);
        }
        while (i <= mid) temp.push_back(nums[i++]);
        while (k <= right) temp.push_back(nums[k++]);
        for (int i = 0; i < temp.size(); i++) nums[left + i] = temp[i];
        return count;
    }

    int reversePairs(vector<int>& nums) {
        return mergeSort(nums, 0, nums.size() - 1);
    }
};
```

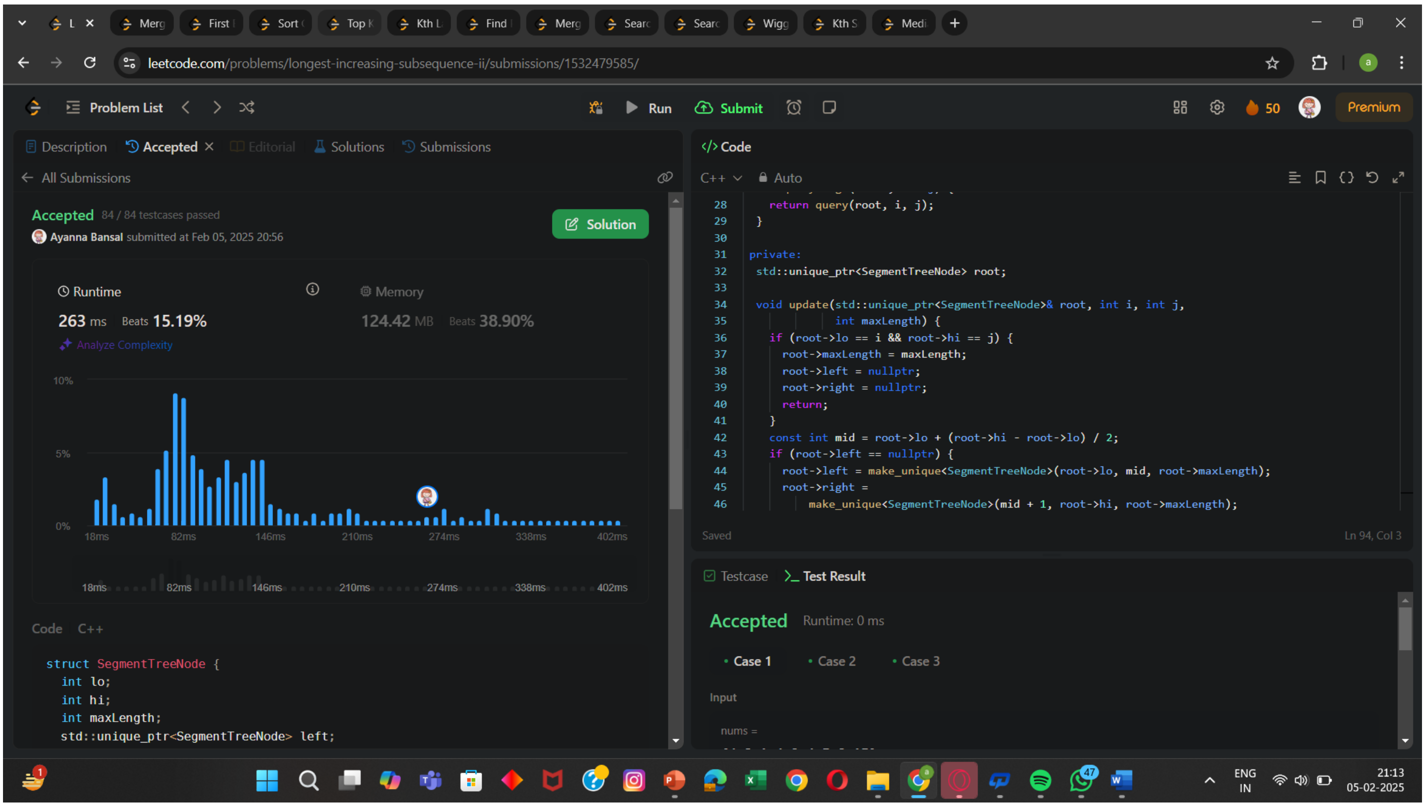
Saved Ln 1, Col 1

nums =

[1,3,2,3,1]

21:17
05-02-2025

2407.Longest Increasing Subsequence II



Accepted 84 / 84 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 20:56

Runtime 263 ms Beats 15.19%

Memory 124.42 MB Beats 38.90%

Code C++

```
struct SegmentTreeNode {
  int lo;
  int hi;
  int maxLength;
  std::unique_ptr<SegmentTreeNode> left;
```

```
    return query(root, i, j);
  }

private:
  std::unique_ptr<SegmentTreeNode> root;

  void update(std::unique_ptr<SegmentTreeNode>& root, int i, int j,
              int maxLength) {
    if (root->lo == i && root->hi == j) {
      root->maxLength = maxLength;
      root->left = nullptr;
      root->right = nullptr;
      return;
    }
    const int mid = root->lo + (root->hi - root->lo) / 2;
    if (root->left == nullptr) {
      root->left = make_unique<SegmentTreeNode>(root->lo, mid, root->maxLength);
      root->right =
          make_unique<SegmentTreeNode>(mid + 1, root->hi, root->maxLength);
```

Saved Ln 94, Col 3

Accepted Runtime: 0 ms

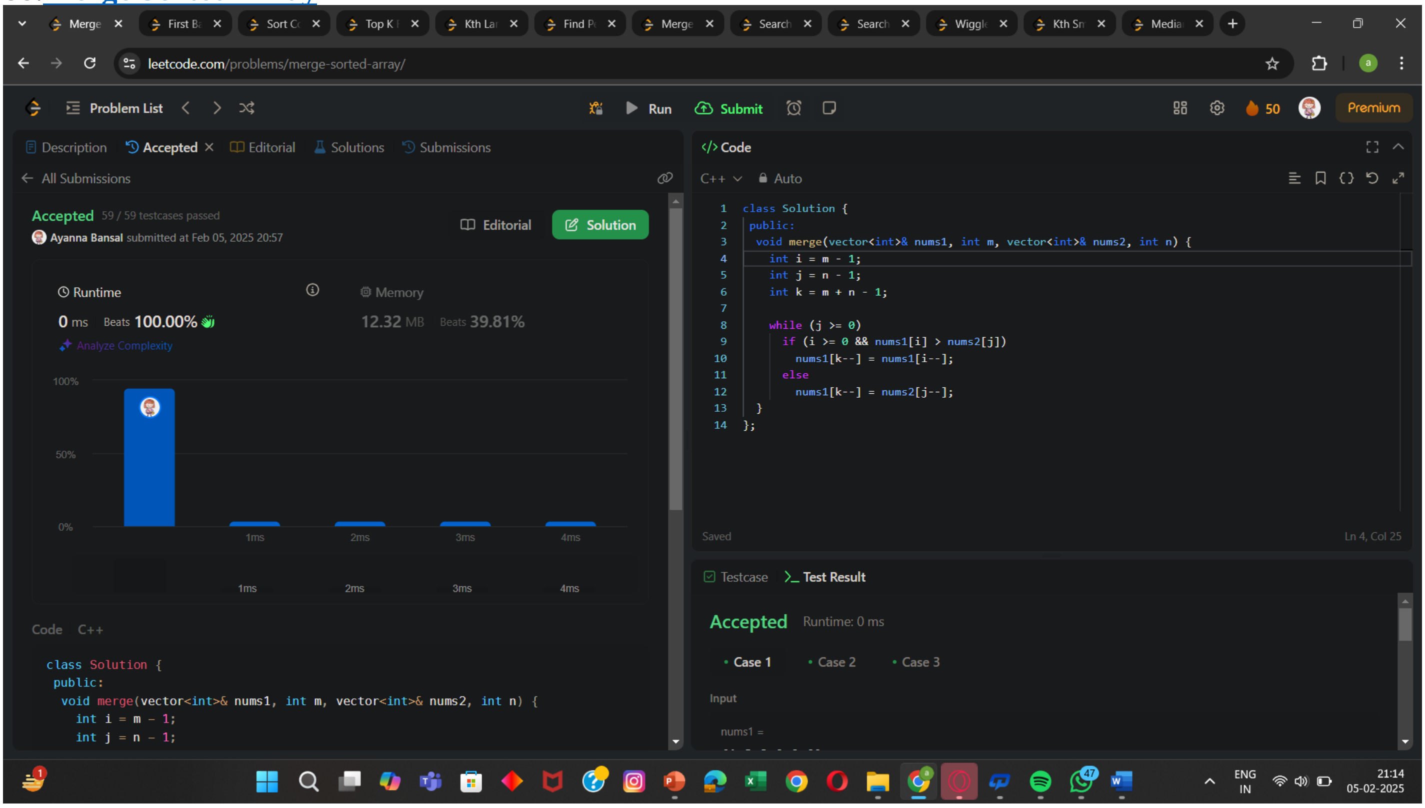
Case 1 Case 2 Case 3

Input

nums =

21:13
05-02-2025

88.Merge Sorted Array



leetcode.com/problems/merge-sorted-array/

Accepted 59 / 59 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 20:57

Runtime 0 ms Beats 100.00%

Memory 12.32 MB Beats 39.81%

Code C++

```
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)
      if (i >= 0 && nums1[i] > nums2[j])
        nums1[k--] = nums1[i--];
      else
        nums1[k--] = nums2[j--];
  }
};
```

Saved Ln 4, Col 25

Testcase Test Result

Accepted Runtime: 0 ms

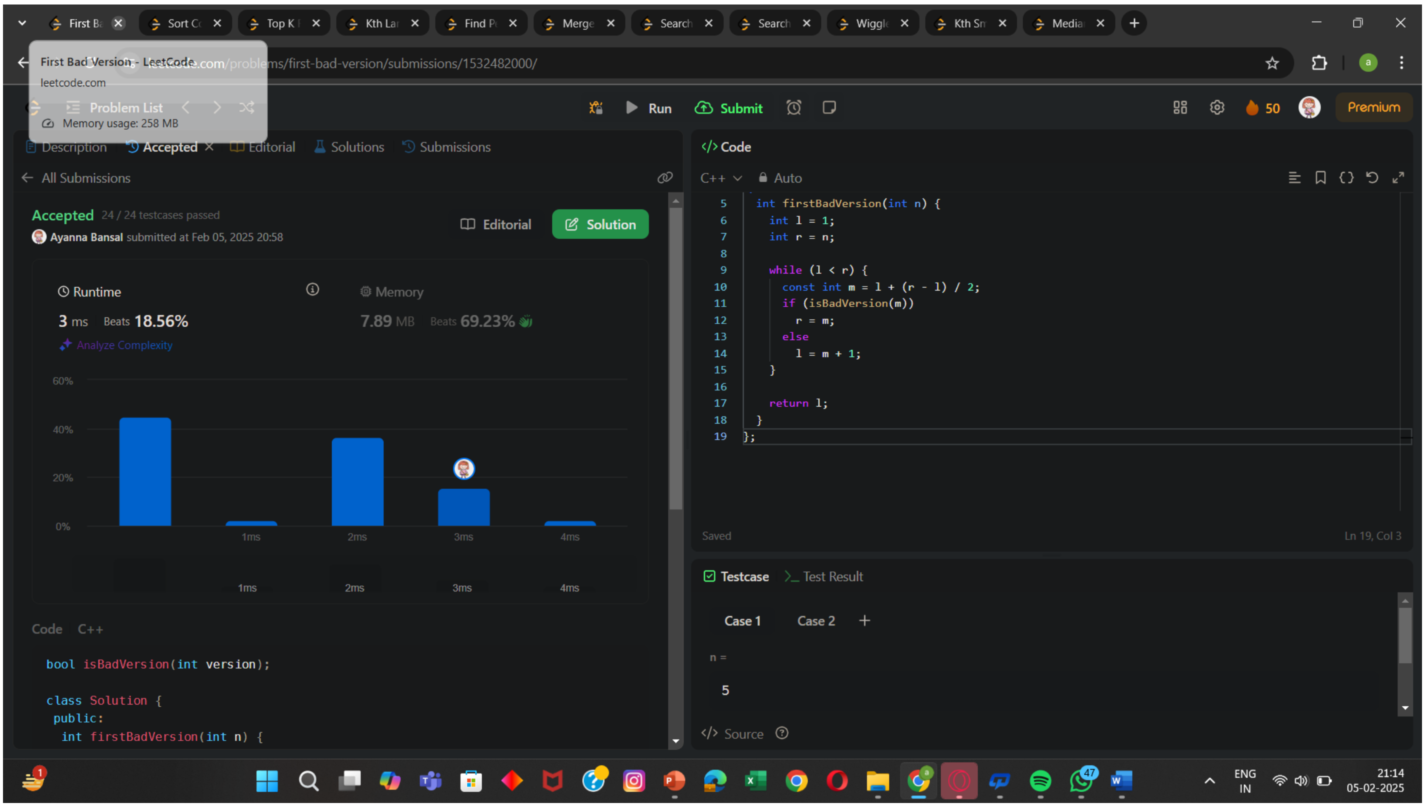
Case 1 Case 2 Case 3

Input

nums1 =

21:14 05-02-2025

278.First Bad Version



leetcode.com/problems/first-bad-version/submissions/1532482000/

First Bad Version - LeetCode
leetcode.com
Memory usage: 258 MB

Accepted 24 / 24 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 20:58

Runtime 3 ms Beats 18.56%

Memory 7.89 MB Beats 69.23%

Code C++

```
bool isBadVersion(int version);

class Solution {
 public:
  int firstBadVersion(int n) {
```

```
  int firstBadVersion(int n) {
    int l = 1;
    int r = n;

    while (l < r) {
      const int m = l + (r - l) / 2;
      if (isBadVersion(m))
        r = m;
      else
        l = m + 1;
    }

    return l;
  }
};
```

Saved Ln 19, Col 3

Testcase Test Result

Case 1 Case 2 +

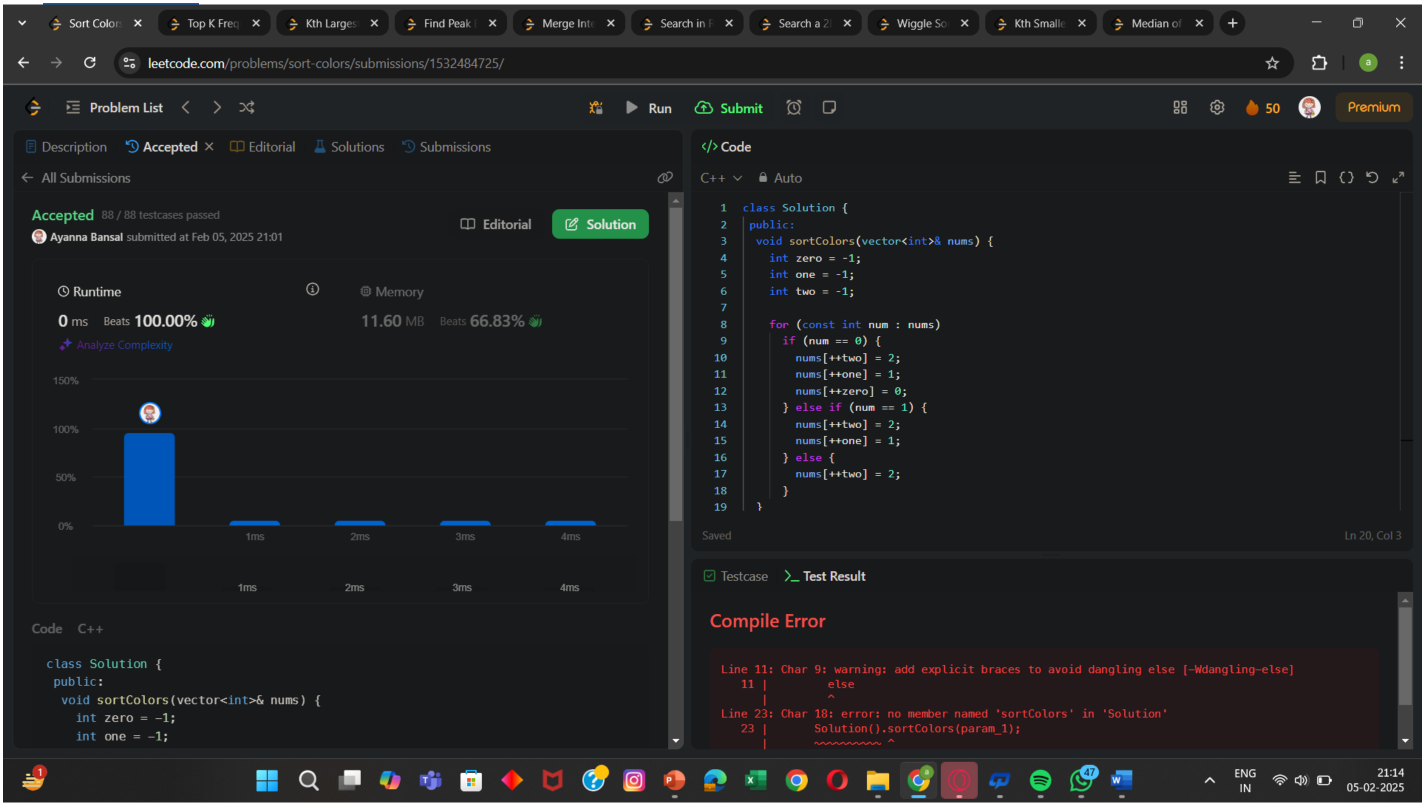
n =

5

Source

21:14 05-02-2025

75.Sort Colors



leetcode.com/problems/sort-colors/submissions/1532484725/

Accepted 88 / 88 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:01

Runtime 0 ms Beats 100.00%

Memory 11.60 MB Beats 66.83%

Code C++

```
class Solution {
 public:
  void sortColors(vector<int>& nums) {
    int zero = -1;
    int one = -1;
```

C++ Auto

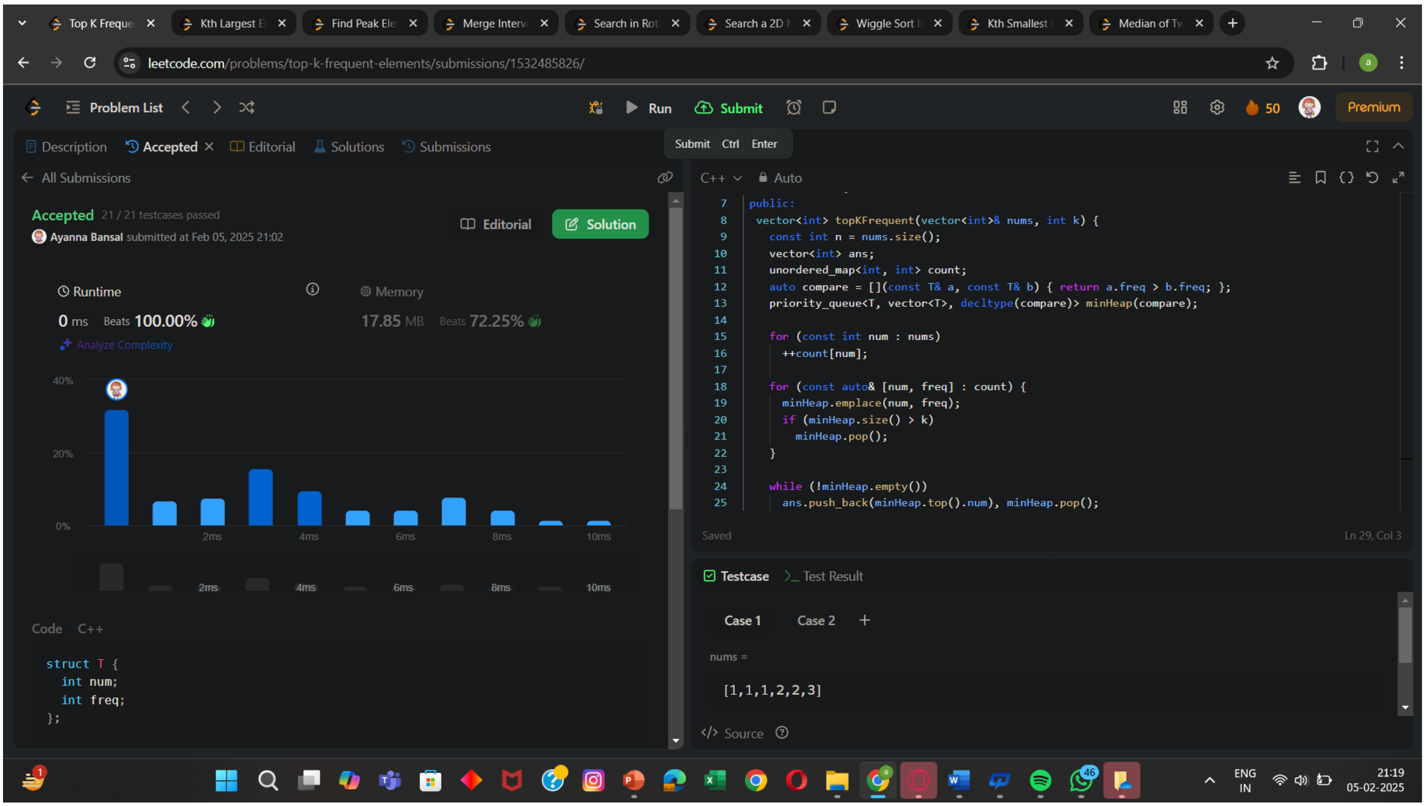
```
class Solution {
 public:
  void sortColors(vector<int>& nums) {
    int zero = -1;
    int one = -1;
    int two = -1;

    for (const int num : nums)
      if (num == 0) {
        nums[++two] = 2;
        nums[++one] = 1;
        nums[++zero] = 0;
      } else if (num == 1) {
        nums[++two] = 2;
        nums[++one] = 1;
      } else {
        nums[++two] = 2;
      }
  }
```

Saved Ln 20, Col 3

Compile Error

```
Line 11: Char 9: warning: add explicit braces to avoid dangling else [-Wdangling-else]
   11 |         else
      |         ^
Line 23: Char 18: error: no member named 'sortColors' in 'Solution'
   23 |       Solution().sortColors(param_1);
      |       ~~~~~~~~~~ ^
```

347.Top K Frequent Elements



leetcode.com/problems/top-k-frequent-elements/submissions/1532485826/

Accepted 21 / 21 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:02

Runtime 0 ms Beats 100.00%

Memory 17.85 MB Beats 72.25%

Code C++

```
struct T {
  int num;
  int freq;
};
```

C++ Auto

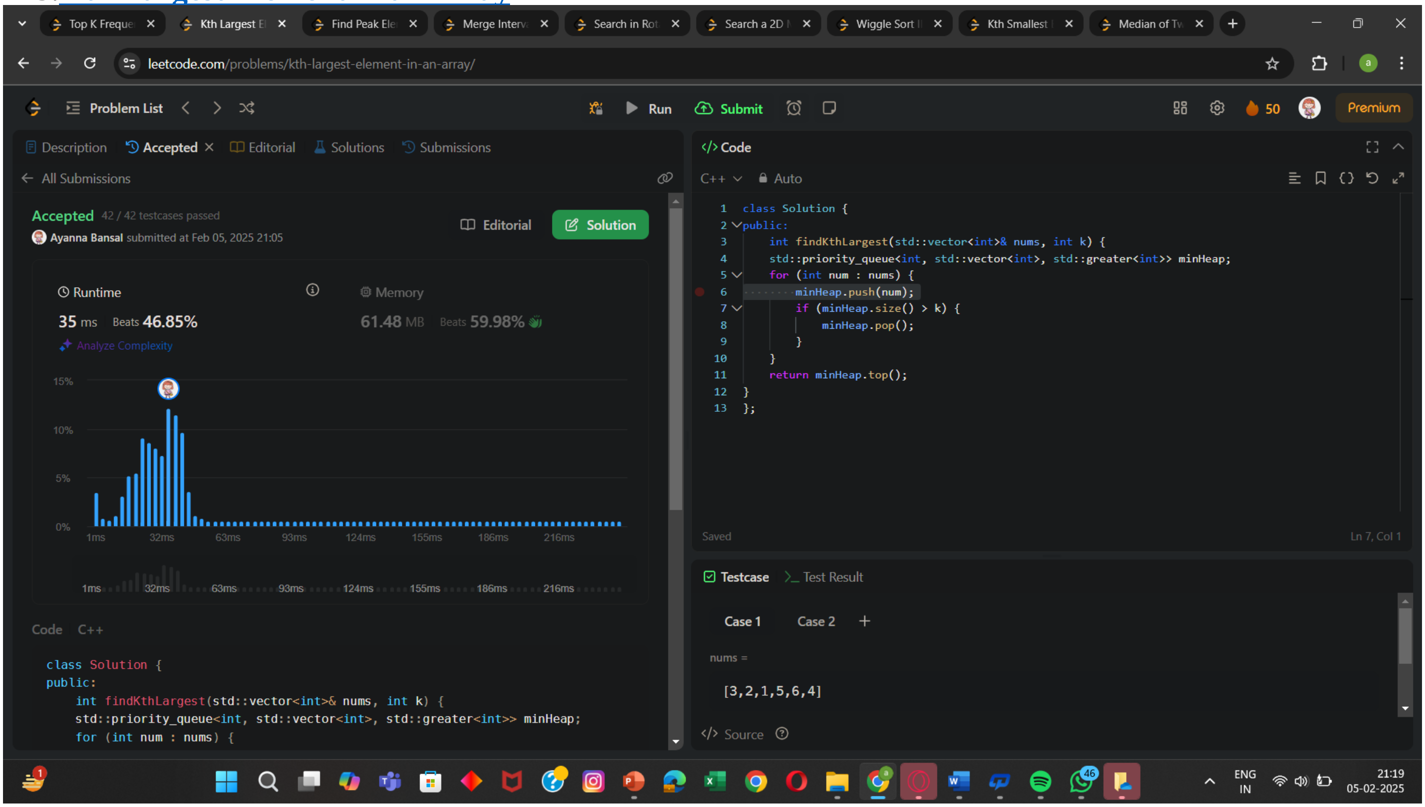
```
 public:
  vector<int> topKFrequent(vector<int>& nums, int k) {
    const int n = nums.size();
    vector<int> ans;
    unordered_map<int, int> count;
    auto compare = [](const T& a, const T& b) { return a.freq > b.freq; };
    priority_queue<T, vector<T>, decltype(compare)> minHeap(compare);

    for (const int num : nums)
      ++count[num];

    for (const auto& [num, freq] : count) {
      minHeap.emplace(num, freq);
      if (minHeap.size() > k)
        minHeap.pop();
    }

    while (!minHeap.empty())
      ans.push_back(minHeap.top().num), minHeap.pop();
```

Saved Ln 29, Col 3

nums =

[1,1,1,2,2,3]

215.Kth Largest Element in an Array



Accepted 42 / 42 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:05

Runtime 35 ms Beats 46.85%

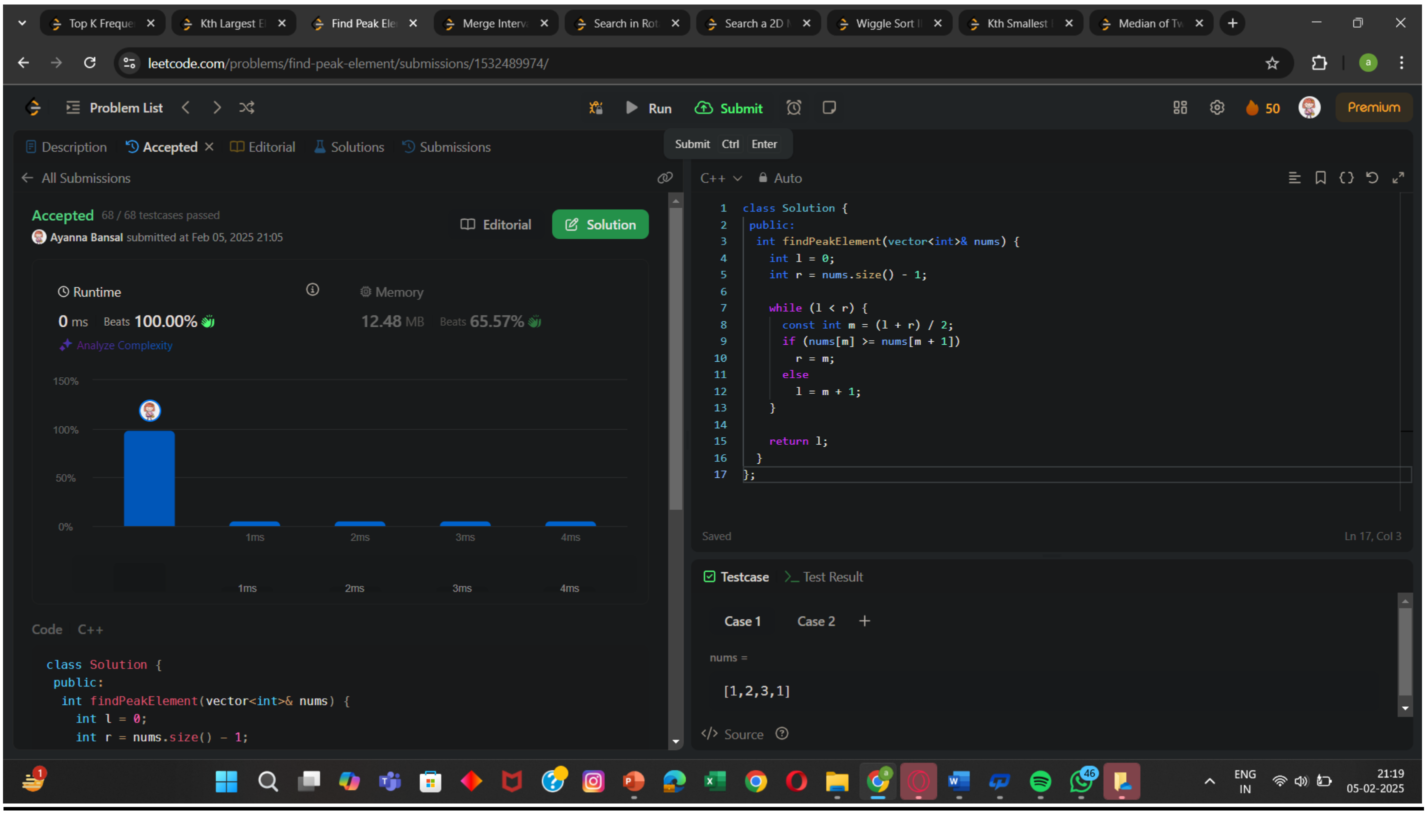
Memory 61.48 MB Beats 59.98%

```cpp
class Solution {
public:
    int findKthLargest(std::vector<int>& nums, int k) {
    std::priority_queue<int, std::vector<int>, std::greater<int>> minHeap;
    for (int num : nums) {
        minHeap.push(num);
        if (minHeap.size() > k) {
            minHeap.pop();
        }
    }
    return minHeap.top();
}
};
```

nums =

```
[3,2,1,5,6,4]
```

162.Find Peak Element



Accepted 68 / 68 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:05

Runtime 0 ms Beats 100.00%

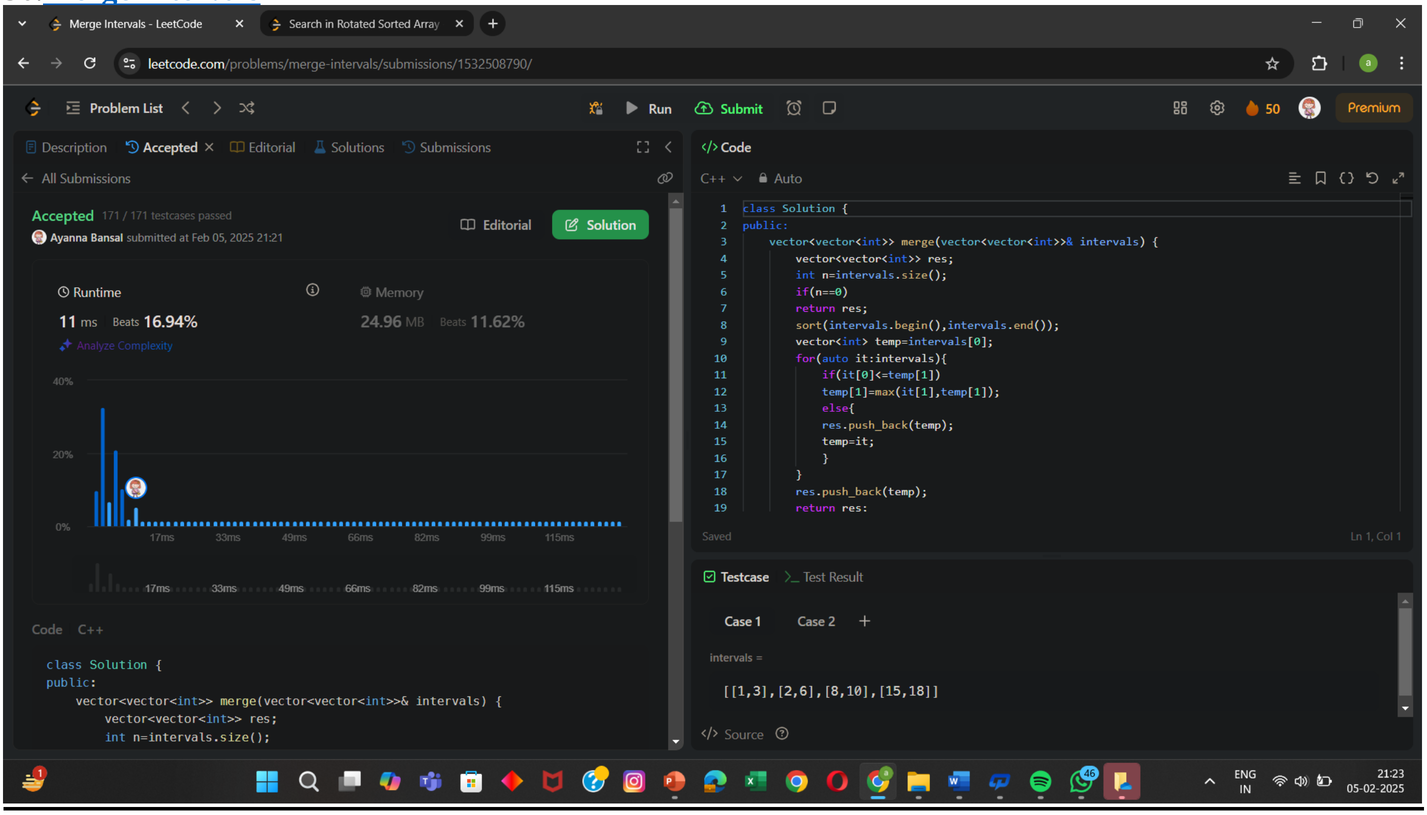
Memory 12.48 MB Beats 65.57%

```cpp
class Solution {
 public:
  int findPeakElement(vector<int>& nums) {
    int l = 0;
    int r = nums.size() - 1;

    while (l < r) {
      const int m = (l + r) / 2;
      if (nums[m] >= nums[m + 1])
        r = m;
      else
        l = m + 1;
    }

    return l;
  }
};
```

nums =

```
[1,2,3,1]
```

56.Merge Intervals



Accepted 171 / 171 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:21

Runtime 11 ms Beats 16.94%

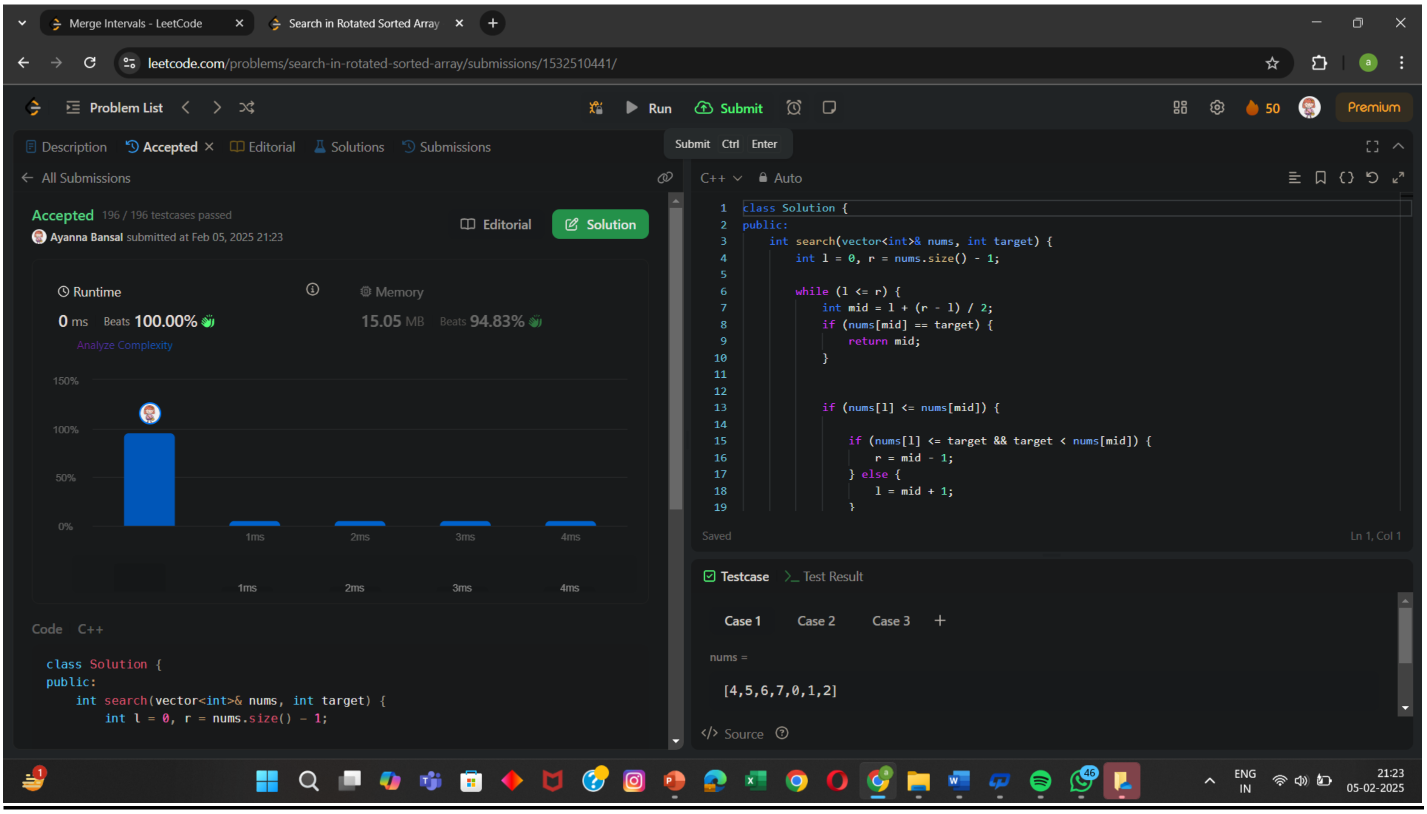
Memory 24.96 MB Beats 11.62%

```cpp
class Solution {
public:
    vector<vector<int>> merge(vector<vector<int>>& intervals) {
        vector<vector<int>> res;
        int n=intervals.size();
        if(n==0)
        return res;
        sort(intervals.begin(),intervals.end());
        vector<int> temp=intervals[0];
        for(auto it:intervals){
            if(it[0]<=temp[1])
            temp[1]=max(it[1],temp[1]);
            else{
            res.push_back(temp);
            temp=it;
            }
        }
        res.push_back(temp);
        return res:
```

intervals =

```
[[1,3],[2,6],[8,10],[15,18]]
```

33.Search in Rotated Sorted Array



Accepted 196 / 196 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:23

Runtime 0 ms Beats 100.00%

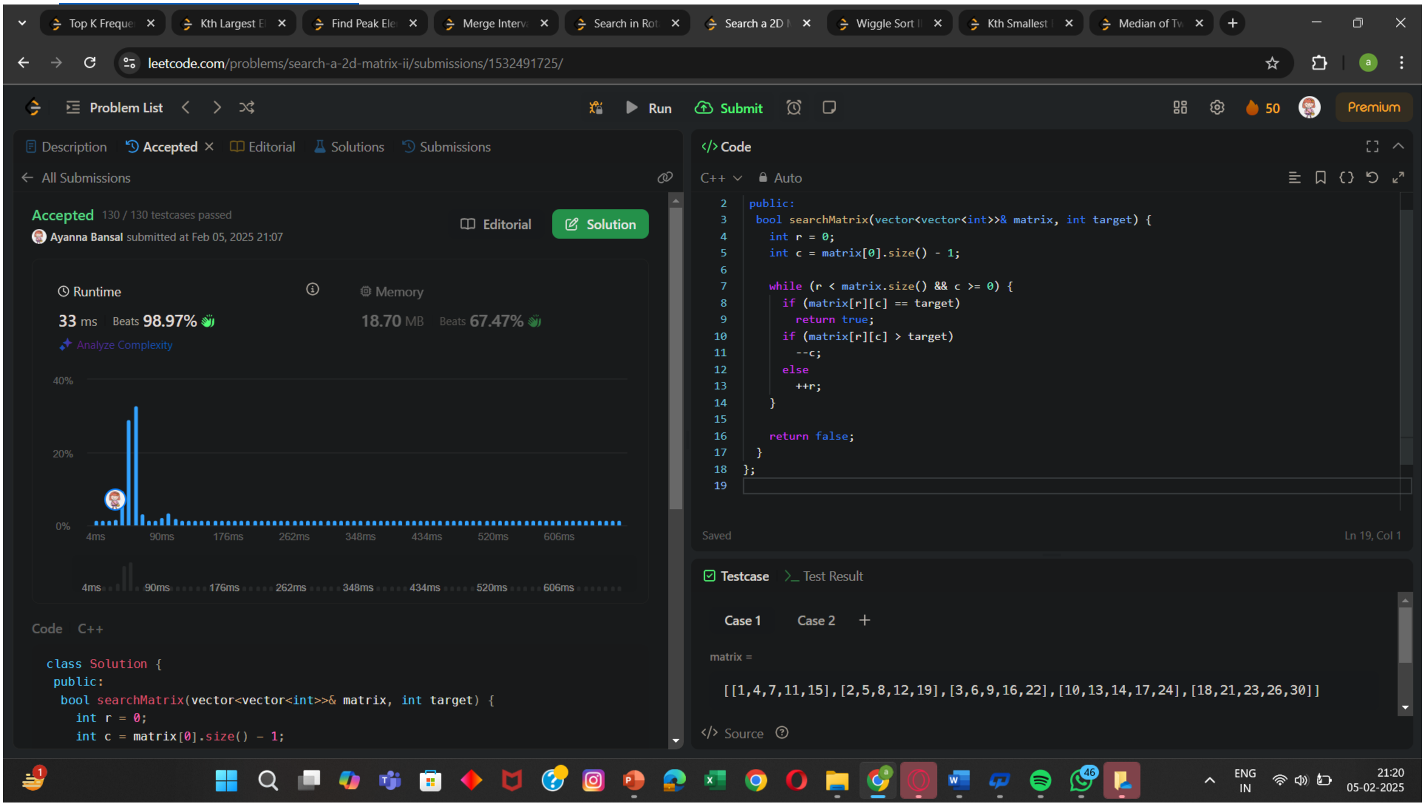
Memory 15.05 MB Beats 94.83%

```cpp
class Solution {
public:
    int search(vector<int>& nums, int target) {
        int l = 0, r = nums.size() - 1;

        while (l <= r) {
            int mid = l + (r - l) / 2;
            if (nums[mid] == target) {
                return mid;
            }


            if (nums[l] <= nums[mid]) {

                if (nums[l] <= target && target < nums[mid]) {
                    r = mid - 1;
                } else {
                    l = mid + 1;
                }
```

nums =

```
[4,5,6,7,0,1,2]
```

240.Search a 2D Matrix II



leetcode.com/problems/search-a-2d-matrix-ii/submissions/1532491725/

Accepted 130 / 130 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:07

Runtime 33 ms Beats 98.97%

Memory 18.70 MB Beats 67.47%

Code C++

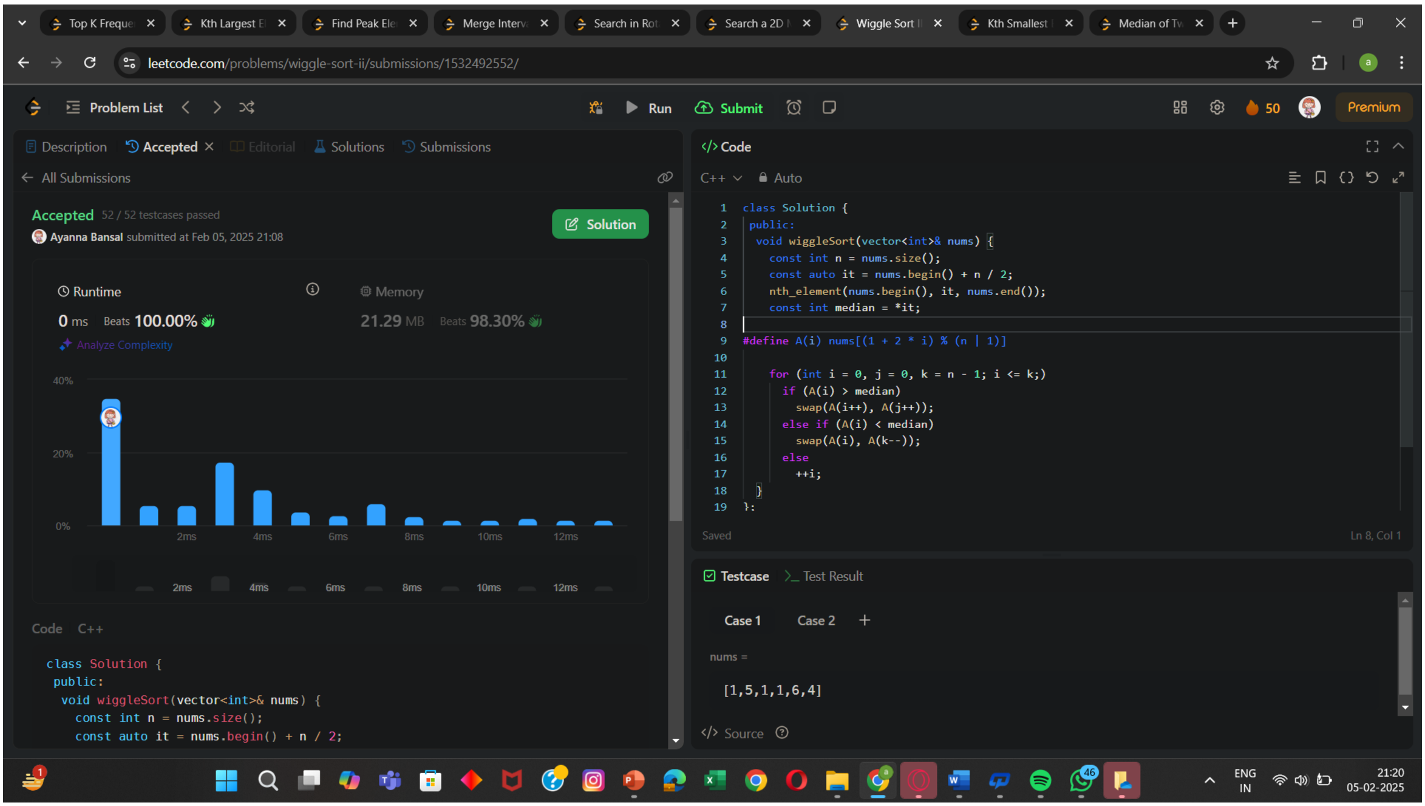
```
class Solution {
 public:
  bool searchMatrix(vector<vector<int>>& matrix, int target) {
    int r = 0;
    int c = matrix[0].size() - 1;
```

```
public:
  bool searchMatrix(vector<vector<int>>& matrix, int target) {
     int r = 0;
     int c = matrix[0].size() - 1;

     while (r < matrix.size() && c >= 0) {
       if (matrix[r][c] == target)
         return true;
       if (matrix[r][c] > target)
         --c;
       else
         ++r;
     }

     return false;
  }
};
```

matrix =

[[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]]

324.Wiggle Sort II



leetcode.com/problems/wiggle-sort-ii/submissions/1532492552/

Accepted 52 / 52 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:08

Runtime 0 ms Beats 100.00%

Memory 21.29 MB Beats 98.30%

Code C++

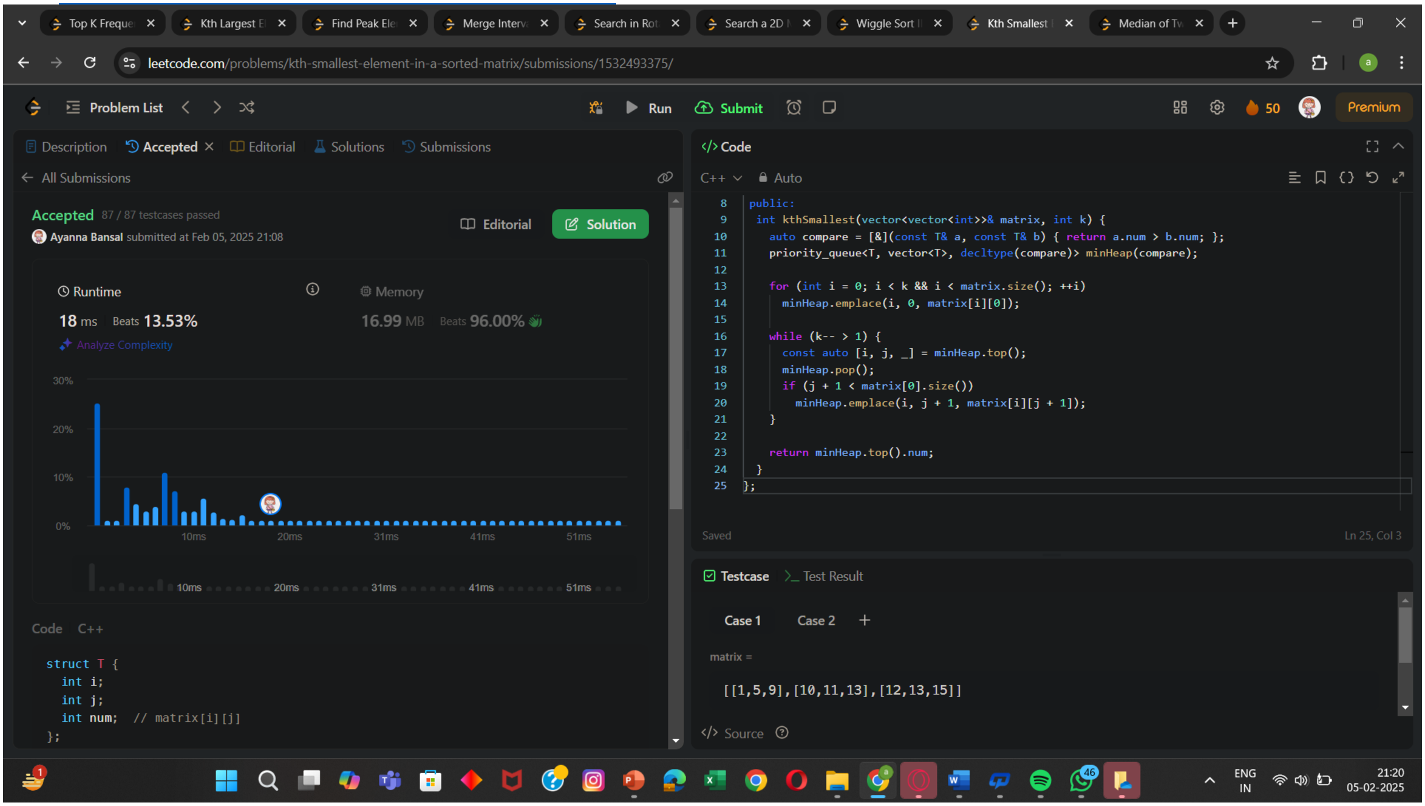
```
class Solution {
 public:
  void wiggleSort(vector<int>& nums) {
    const int n = nums.size();
    const auto it = nums.begin() + n / 2;
```

```
class Solution {
 public:
  void wiggleSort(vector<int>& nums) {
    const int n = nums.size();
    const auto it = nums.begin() + n / 2;
    nth_element(nums.begin(), it, nums.end());
    const int median = *it;

#define A(i) nums[(1 + 2 * i) % (n | 1)]

    for (int i = 0, j = 0, k = n - 1; i <= k;)
      if (A(i) > median)
        swap(A(i++), A(j++));
      else if (A(i) < median)
        swap(A(i), A(k--));
      else
        ++i;
  }
};
```

nums =

[1,5,1,1,6,4]

378.Kth Smallest Element in a Sorted Matrix



leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix/submissions/1532493375/

Accepted 87 / 87 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:08

Runtime 18 ms Beats 13.53%

Memory 16.99 MB Beats 96.00%

Code C++

```cpp
struct T {
  int i;
  int j;
  int num;  // matrix[i][j]
};
```

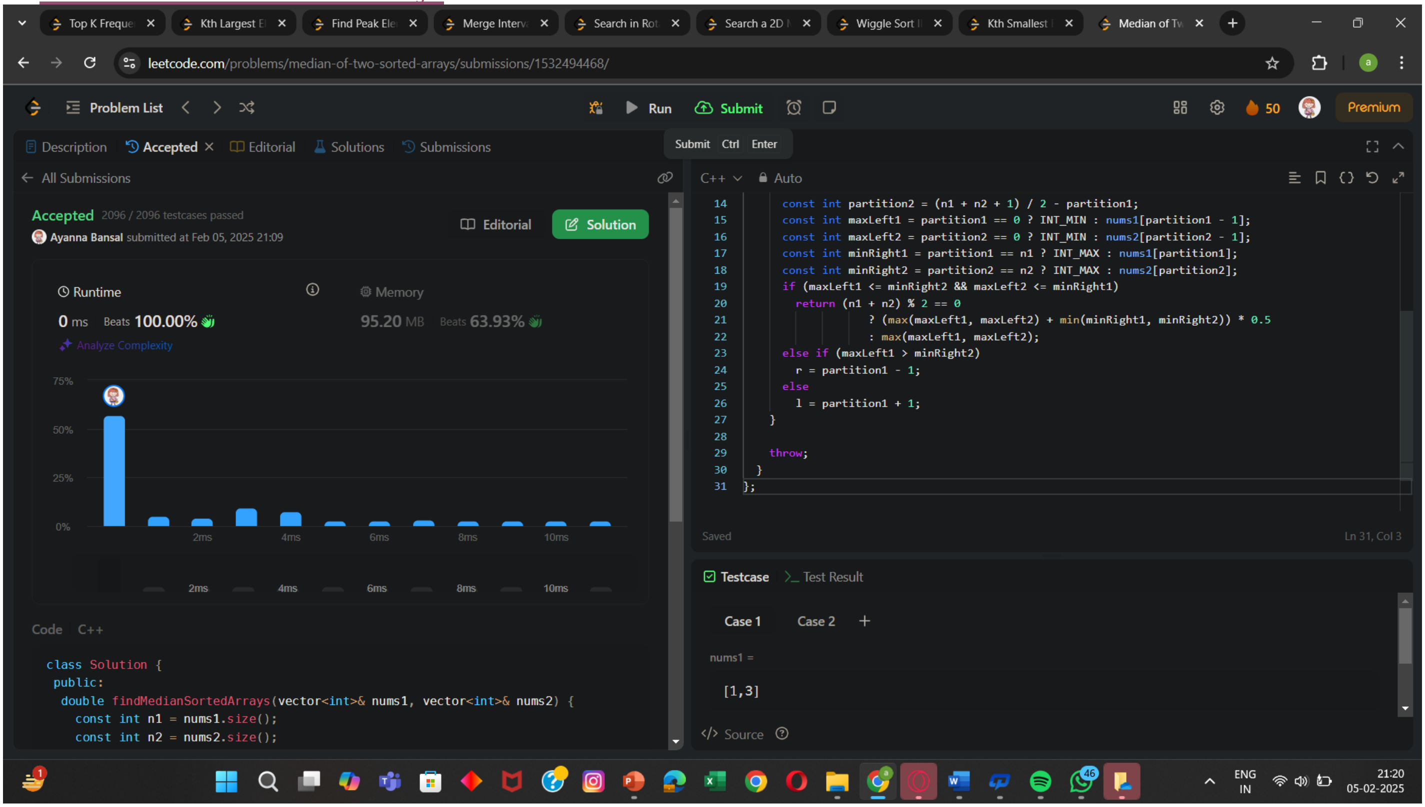
```cpp
public:
 int kthSmallest(vector<vector<int>>& matrix, int k) {
   auto compare = [&](const T& a, const T& b) { return a.num > b.num; };
   priority_queue<T, vector<T>, decltype(compare)> minHeap(compare);

   for (int i = 0; i < k && i < matrix.size(); ++i)
     minHeap.emplace(i, 0, matrix[i][0]);

   while (k-- > 1) {
     const auto [i, j, _] = minHeap.top();
     minHeap.pop();
     if (j + 1 < matrix[0].size())
       minHeap.emplace(i, j + 1, matrix[i][j + 1]);
   }

   return minHeap.top().num;
 }
};
```

Saved Ln 25, Col 3

matrix =

```
[[1,5,9],[10,11,13],[12,13,15]]
```

4.Median of Two Sorted Arrays



leetcode.com/problems/median-of-two-sorted-arrays/submissions/1532494468/

Accepted 2096 / 2096 testcases passed

Ayanna Bansal submitted at Feb 05, 2025 21:09

Runtime 0 ms Beats 100.00%

Memory 95.20 MB Beats 63.93%

Code C++

```cpp
class Solution {
 public:
  double findMedianSortedArrays(vector<int>& nums1, vector<int>& nums2) {
    const int n1 = nums1.size();
    const int n2 = nums2.size();
```

```cpp
      const int partition2 = (n1 + n2 + 1) / 2 - partition1;
      const int maxLeft1 = partition1 == 0 ? INT_MIN : nums1[partition1 - 1];
      const int maxLeft2 = partition2 == 0 ? INT_MIN : nums2[partition2 - 1];
      const int minRight1 = partition1 == n1 ? INT_MAX : nums1[partition1];
      const int minRight2 = partition2 == n2 ? INT_MAX : nums2[partition2];
      if (maxLeft1 <= minRight2 && maxLeft2 <= minRight1)
        return (n1 + n2) % 2 == 0
                   ? (max(maxLeft1, maxLeft2) + min(minRight1, minRight2)) * 0.5
                   : max(maxLeft1, maxLeft2);
      else if (maxLeft1 > minRight2)
        r = partition1 - 1;
      else
        l = partition1 + 1;
    }

    throw;
  }
};
```

Saved Ln 31, Col 3

nums1 =

```
[1,3]
```