**Submitted By : Diksha Pandey**

**UID: 22BCS103359**

**SEC: FL-IOT-602-A**

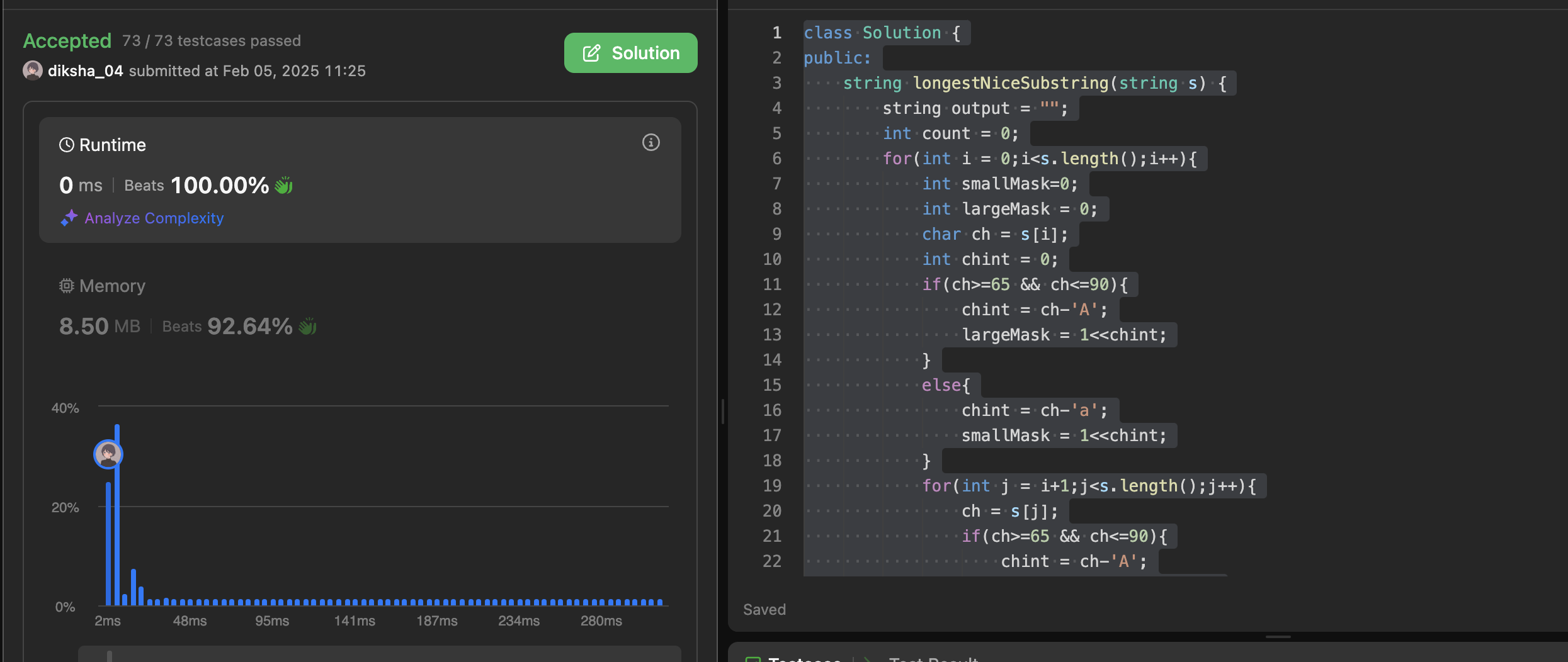
Advanced Programming Assignment

* 1. 1763.[Longest Nice Substring](https://leetcode.com/problems/longest-nice-substring/description/)<br>

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';

largeMask = 1<<chint;

}

else{

chint = ch-'a';

smallMask = 1<<chint;

}

for(int j = i+1;j<s.length();j++){

ch = s[j];

if(ch>=65 && ch<=90){

chint = ch-'A';

largeMask |= 1<<chint;

}

else{

chint = ch-'a';

smallMask |= 1<<chint;

}

//checking for nice

if((smallMask^largeMask) == 0){

if(count<j-i+1){

count = j-i+1;

string temp(s.begin()+i,s.begin()+j+1);

output = temp;

}

}

}

}

return output;

}

};

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

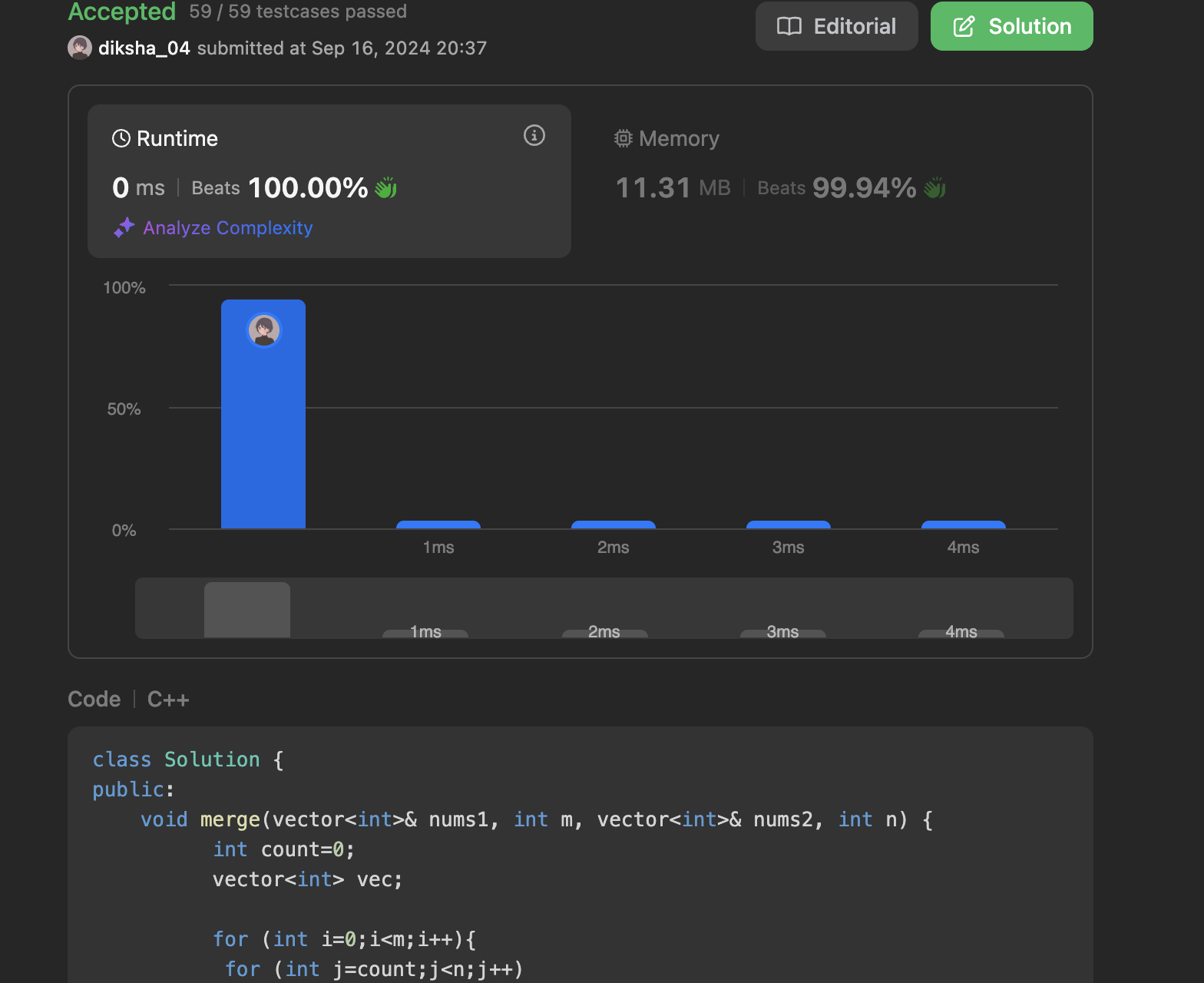
class Solution {

public:

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

int count=0;

vector<int> vec;

for (int i=0;i<m;i++){

for (int j=count;j<n;j++)

{

if (nums1[i]>=nums2[j]){

vec.push\_back(nums2[j]);

count++;

}

}

vec.push\_back(nums1[i]);

}

for (int i =count;i<n;i++){

vec.push\_back(nums2[i]);

}

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

nums1[i]=vec[i];

}

}

};

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

class Solution {

public:

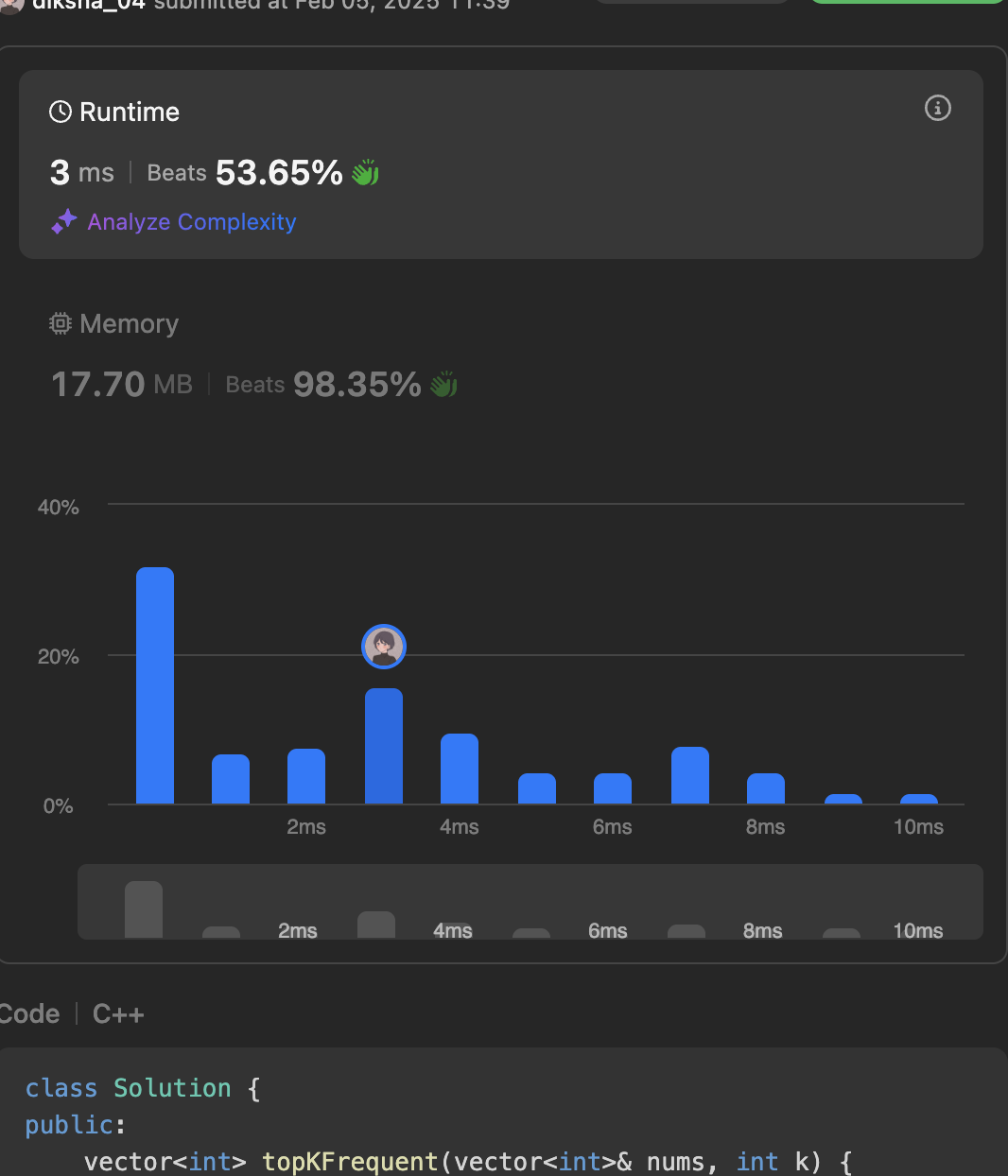
vector<int> topKFrequent(vector<int>& nums, int k) {

unordered\_map<int, int> counter;

for (int n : nums) {

counter[n]++;

}

auto comp = [](pair<int, int>& a, pair<int, int>& b) {

return a.second < b.second;

};

priority\_queue<pair<int, int>, vector<pair<int, int>>, decltype(comp)> heap(comp);

for (auto& entry : counter) {

heap.push({entry.first, entry.second});

}

vector<int> res;

while (k-- > 0) {

res.push\_back(heap.top().first);

heap.pop();

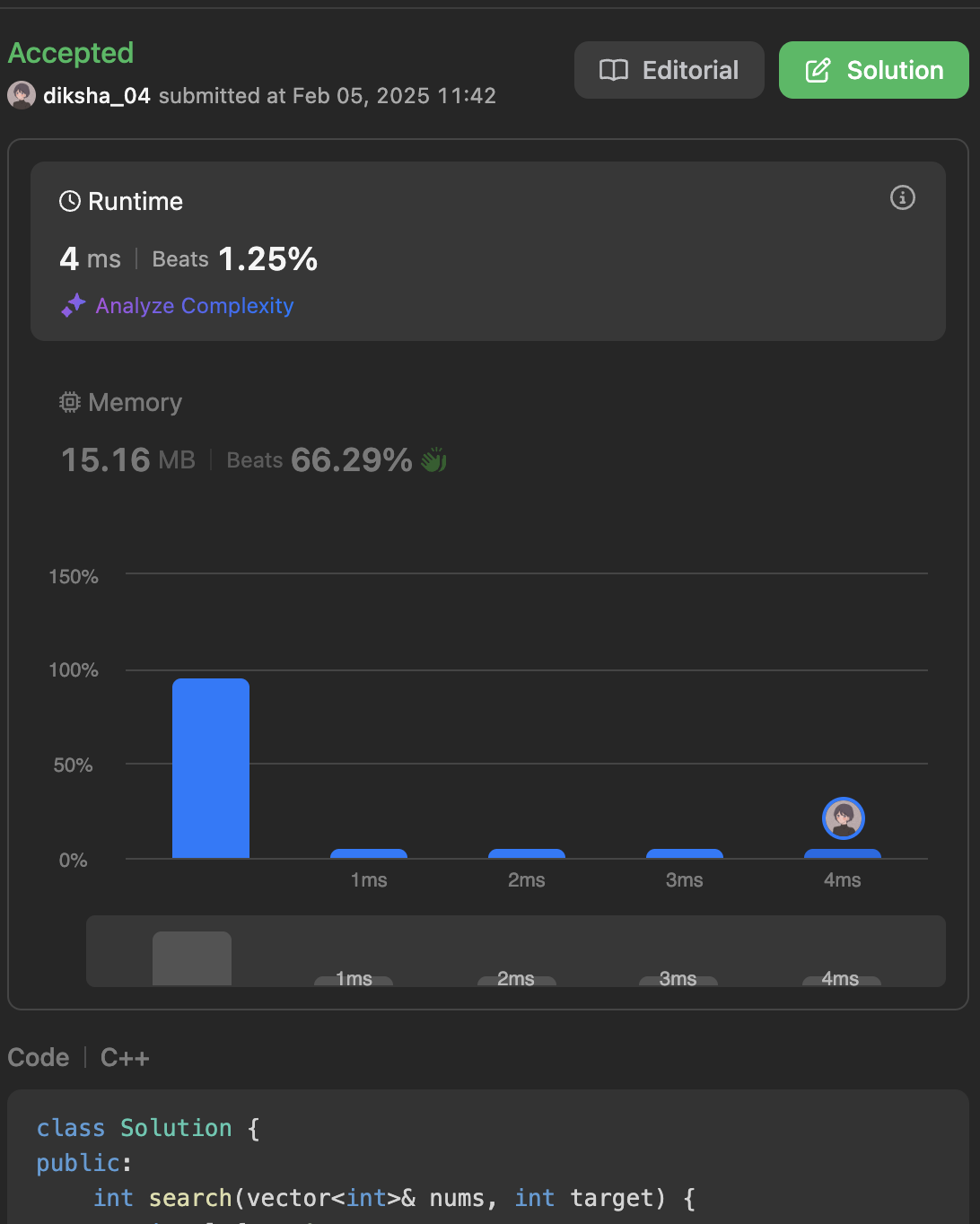
}

return res;

}

};

4. 33.[Search in Rotated Sorted Array](https://leetcode.com/problems/search-in-rotated-sorted-array/description/)<br>



class Solution {

public:

int search(vector<int>& nums, int target) {

int left = 0;

int right = nums.size() - 1;

while (left <= right) {

int mid = (left + right) / 2;

if (nums[mid] == target) {

return mid;

} else if (nums[mid] >= nums[left]) {

if (nums[left] <= target && target <= nums[mid]) {

right = mid - 1;

} else {

left = mid + 1;

}

} else {

if (nums[mid] <= target && target <= nums[right]) {

left = mid + 1;

} else {

right = mid - 1;

}

}

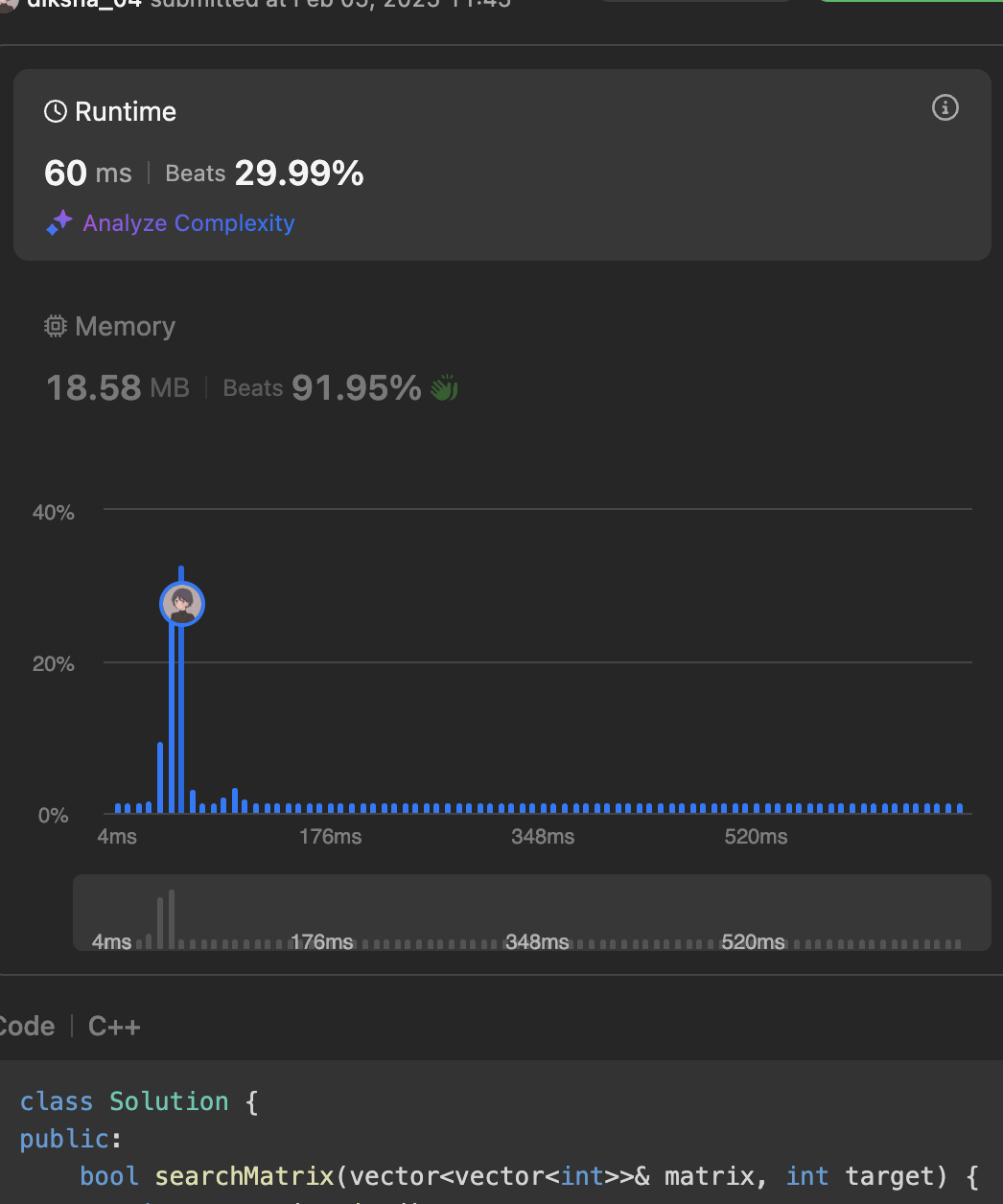
}

return -1;

}

};

5.240.[Search a 2D Matrix II](https://leetcode.com/problems/search-a-2d-matrix-ii/description/)<br>



class Solution {

public:

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

int n=matrix.size();

int m=matrix[0].size();

int row=0,col=m-1;

while(row<n && col>=0){

if(matrix[row][col]==target){

return true;

}else if(matrix[row][col]<target){

row++;

}else{

col--;

}

}

return false;

}

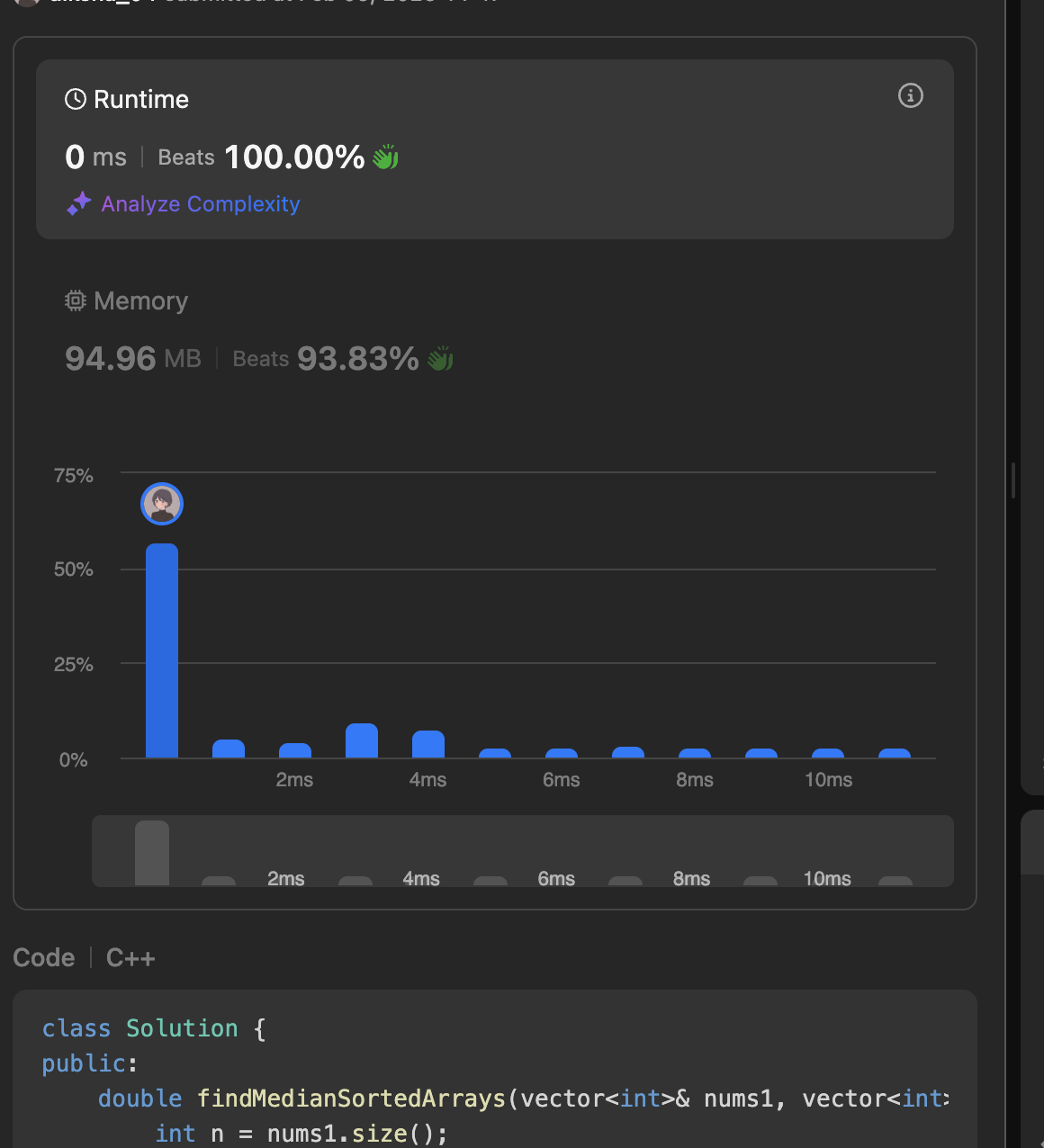
};

6.4.[Median of Two Sorted Arrays](https://leetcode.com/problems/median-of-two-sorted-arrays/description/)<br>

class Solution {

public:

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

int n = nums1.size();

int m = nums2.size();

int i = 0, j = 0, m1 = 0, m2 = 0;

// Find median.

for (int count = 0; count <= (n + m) / 2; count++) {

m2 = m1;

if (i != n && j != m) {

if (nums1[i] > nums2[j]) {

m1 = nums2[j++];

} else {

m1 = nums1[i++];

}

} else if (i < n) {

m1 = nums1[i++];

} else {

m1 = nums2[j++];

}

}

// Check if the sum of n and m is odd.

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

return static\_cast<double>(m1);

} else {

double ans = static\_cast<double>(m1) + static\_cast<double>(m2);

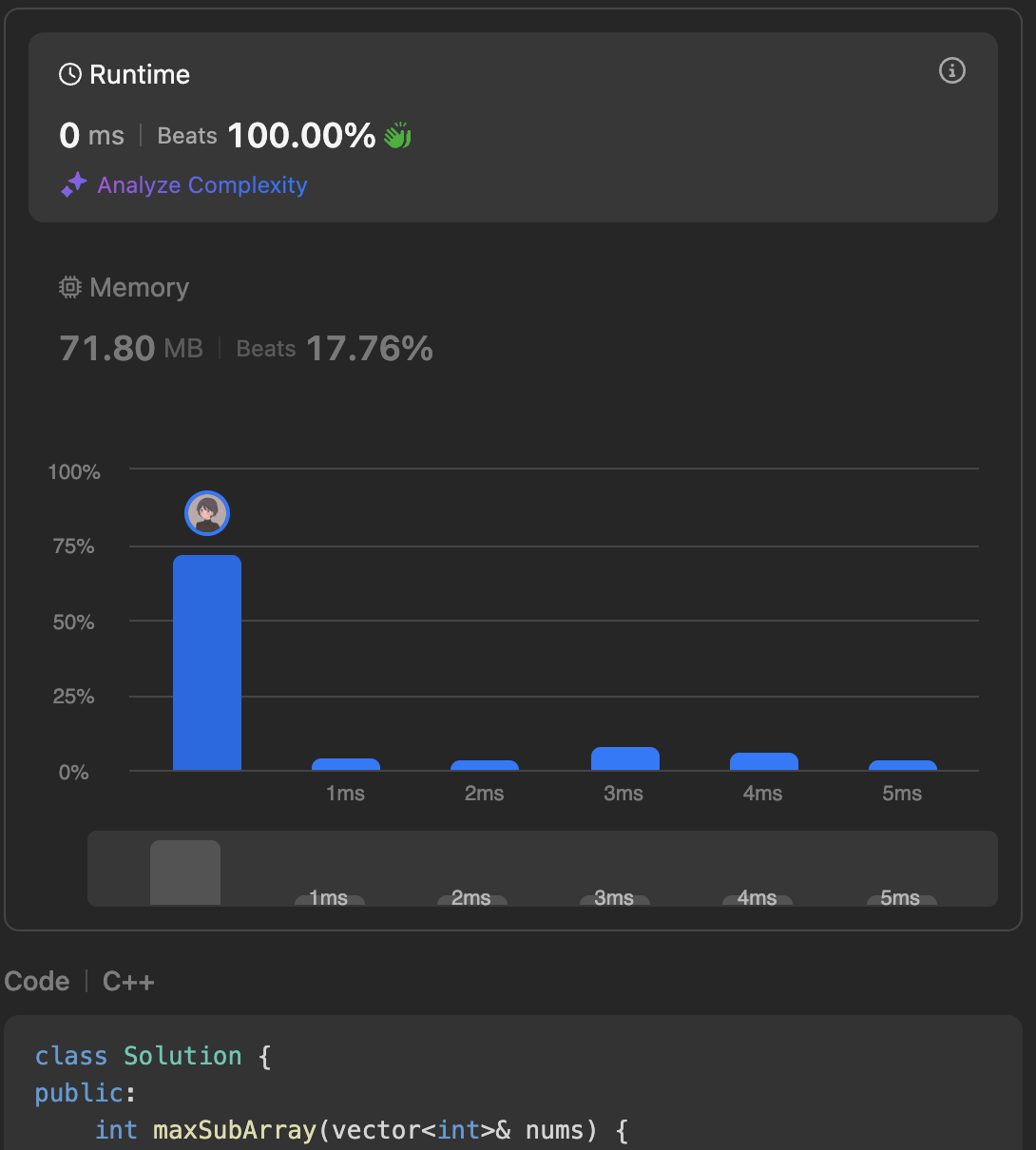
return ans / 2.0;

}

}

};

7.53.[Maximum Subarray](https://leetcode.com/problems/maximum-subarray/description/)<br>



class Solution {

public:

int maxSubArray(vector<int>& nums) {

int res = nums[0];

int total = 0;

for (int n : nums) {

if (total < 0) {

total = 0;

}

total += n;

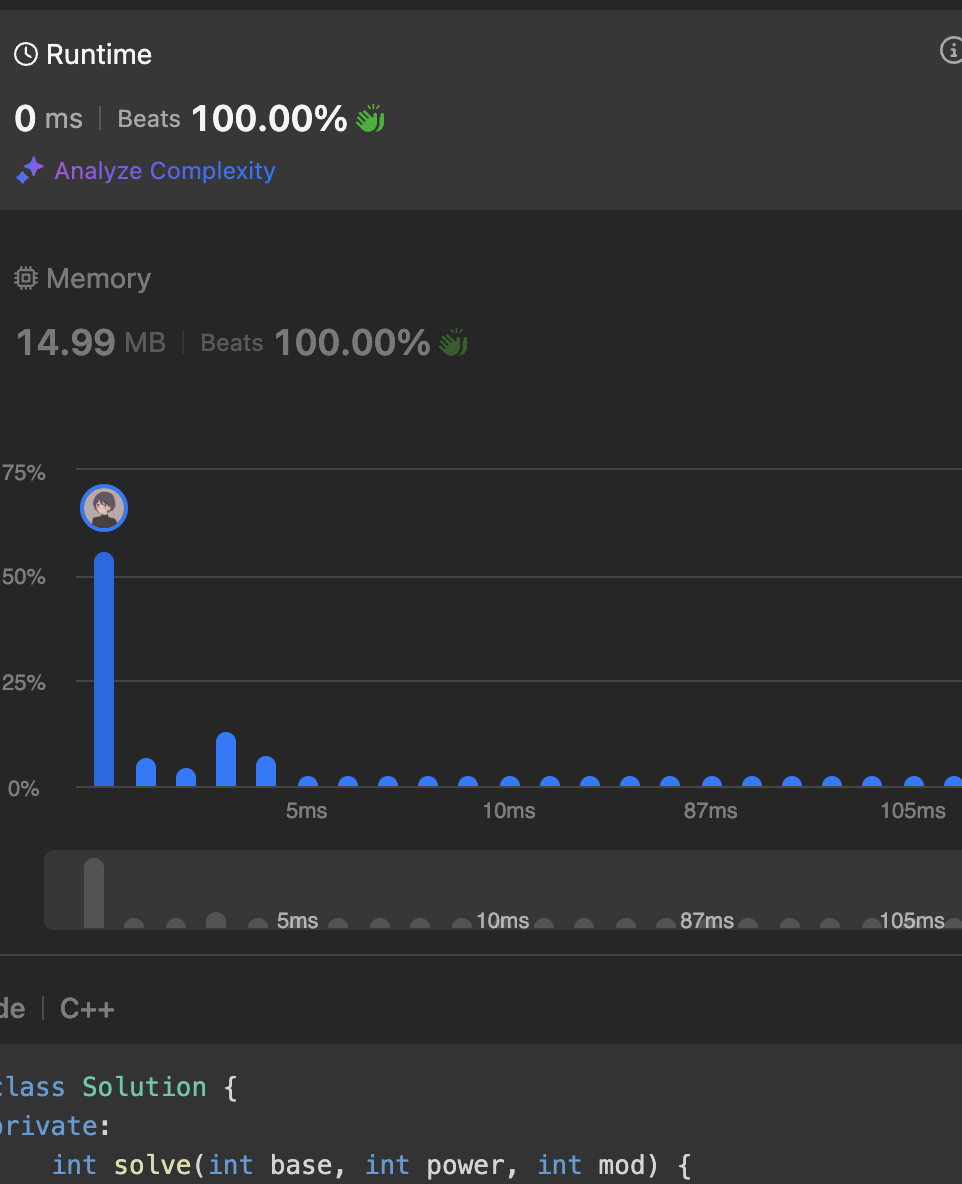
res = max(res, total);

}

return res;

}

};

8.372.[Super Pow](https://leetcode.com/problems/super-pow/description/)<br>

class Solution {

private:

int solve(int base, int power, int mod) {

int ans = 1;

while (power > 0) {

if (power & 1) {

ans = (ans \* base) % mod;

}

base = (base \* base) % mod;

power >>= 1;

}

return ans;

}

public:

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

a%=1337;

int n = b.size();

int m = 1140;

int expi = 0;

for(int i : b){

expi = (expi\*10+i)%m;

}

if (expi == 0) {

expi = m;

}

return solve(a,expi,1337);

}

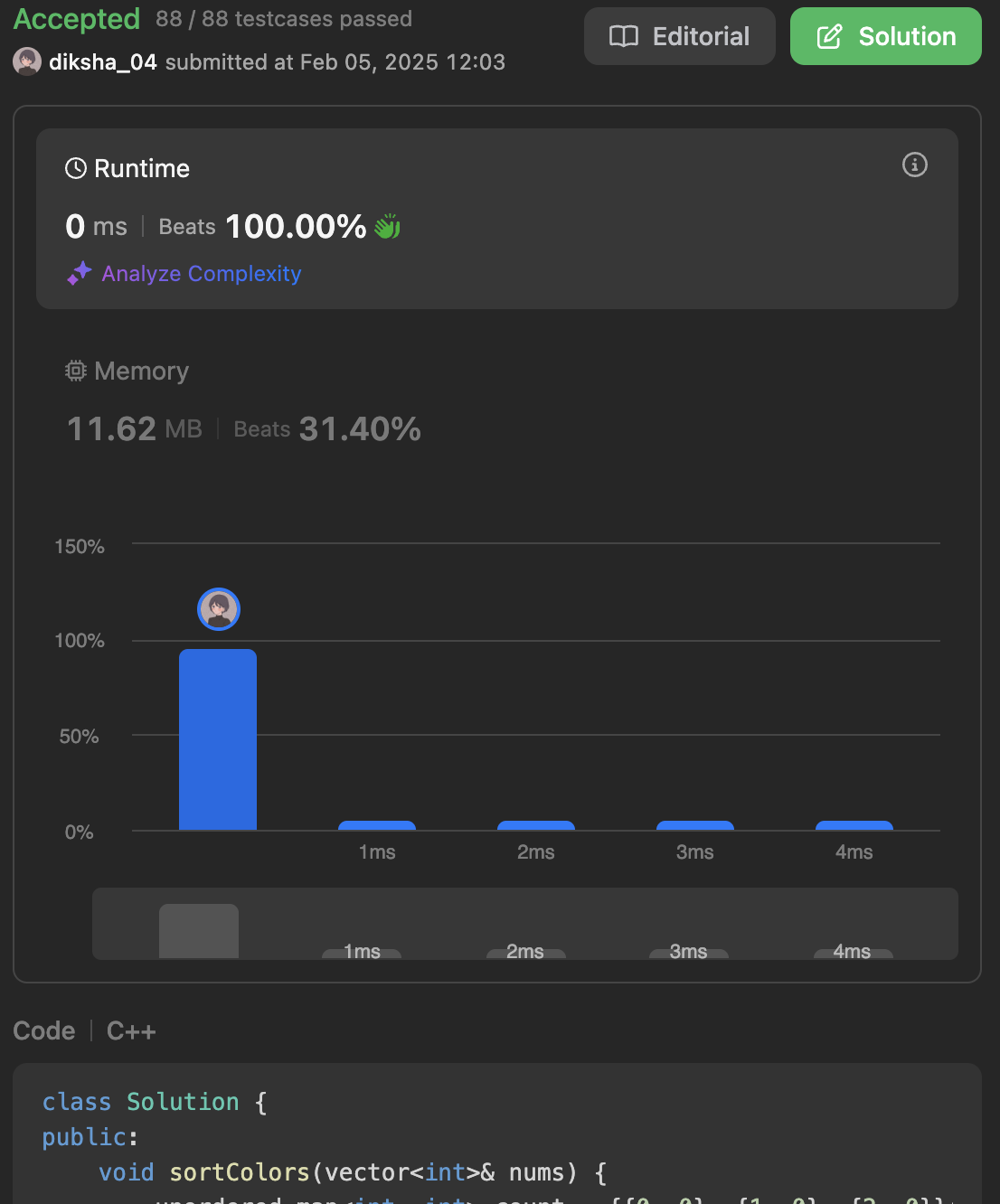
};

9.75.[Sort Colors](https://leetcode.com/problems/sort-colors/description/)<br>

class Solution {

public:

void sortColors(vector<int>& nums) {

unordered\_map<int, int> count = {{0, 0}, {1, 0}, {2, 0}};

for (int num : nums) {

count[num]++;

}

int idx = 0;

for (int color = 0; color < 3; color++) {

int freq = count[color];

for (int j = 0; j < freq; j++) {

nums[idx] = color;

idx++;

}

}

}

};

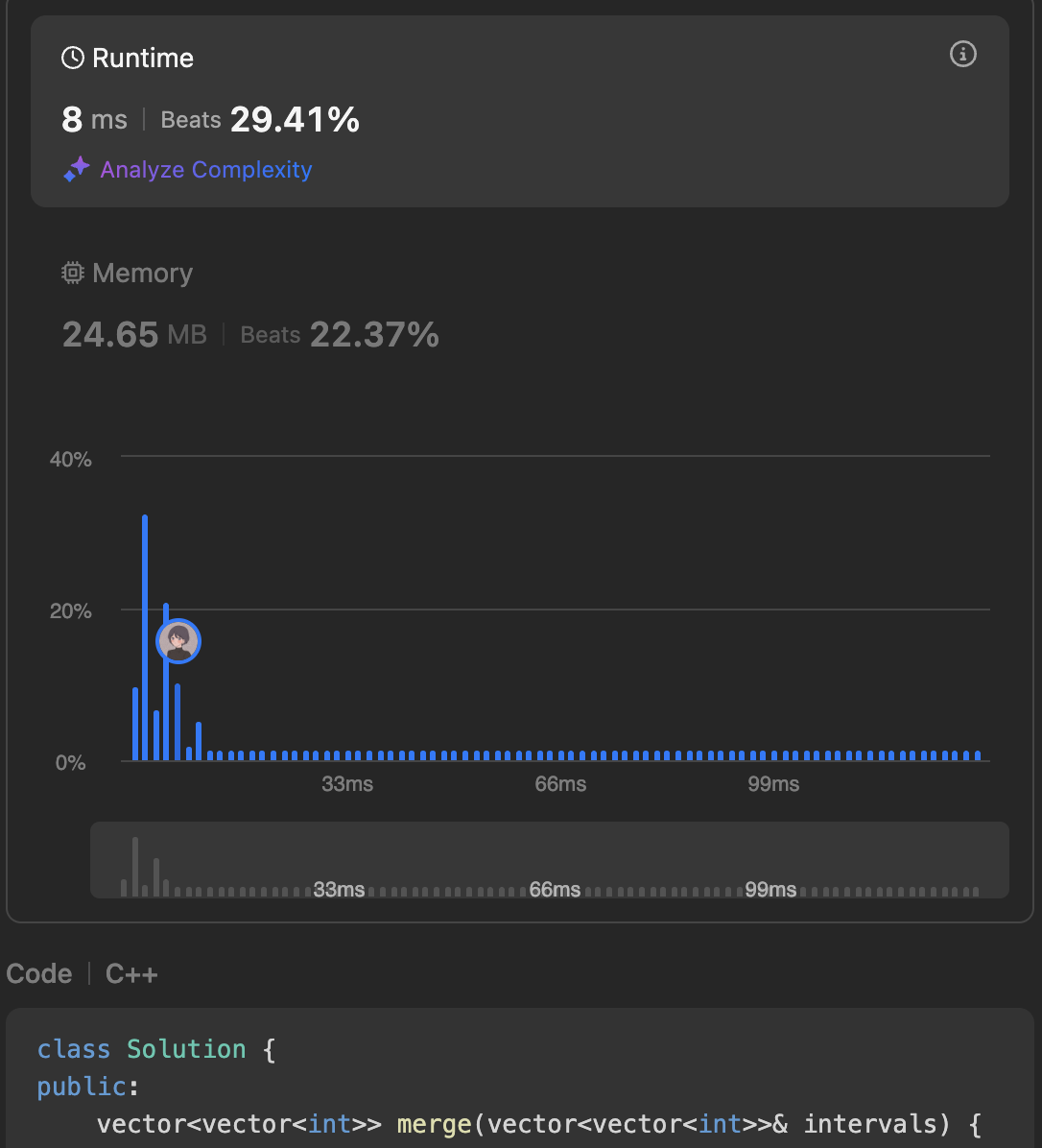
10.56.[Merge Intervals](https://leetcode.com/problems/merge-intervals/description/)<br>

class Solution {

public:

vector<vector<int>> merge(vector<vector<int>>& intervals) {

sort(intervals.begin(), intervals.end(), [](const vector<int>& a, const vector<int>& b) {

return a[0] < b[0];

});

vector<vector<int>> merged;

vector<int> prev = intervals[0];

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

vector<int> interval = intervals[i];

if (interval[0] <= prev[1]) {

prev[1] = max(prev[1], interval[1]);

} else {

merged.push\_back(prev);

prev = interval;

}

}

merged.push\_back(prev);

return merged;

}

};