CU
CHANDIGARH UNIVERSITY

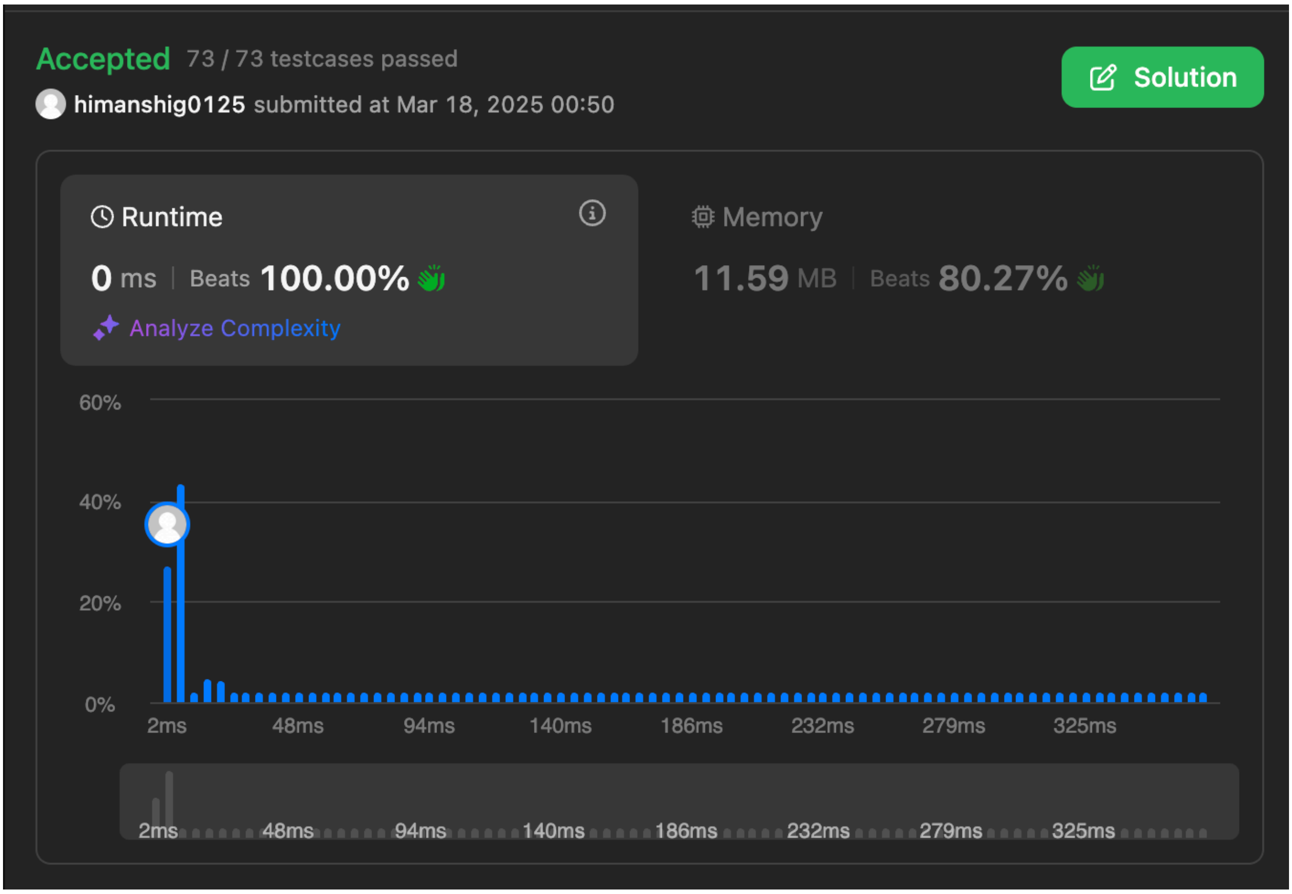
DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
Discover. Learn. Empower.

Name - Himanshi Gupta
Semester - 6
Subject - Advanced Programming - 2

UID - 22CBS15329
Date - 18-03-2025
Subject Code - 22CSP-351

1. Longest Nice Substring

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        int n=s.length();
        if (s.length()<2) {
            return "";
        }
        bool lower[26]={false};
        bool upper[26]={false};
        for(char c:s){
            if(islower(c)){
                lower[c-'a']=true;
            }
            else{
                upper[c-'A']=true;
            }
        }
        for(int i=0;i<n;i++){
            char c=s[i];
            if(islower(c)&&!upper[c-'a']){
                string left=longestNiceSubstring(s.substr(0,i));
                string right=longestNiceSubstring(s.substr(i+1));
                return left.length()>=right.length()?left:right;
            }
            if(isupper(c)&&!lower[c-'A']){
                string left=longestNiceSubstring(s.substr(0,i));
                string right=longestNiceSubstring(s.substr(i+1));
                return left.length()>=right.length()?left:right;
            }
        }
        return s;
    }
};
```



Accepted 73 / 73 testcases passed

himanshig0125 submitted at Mar 18, 2025 00:50

Runtime

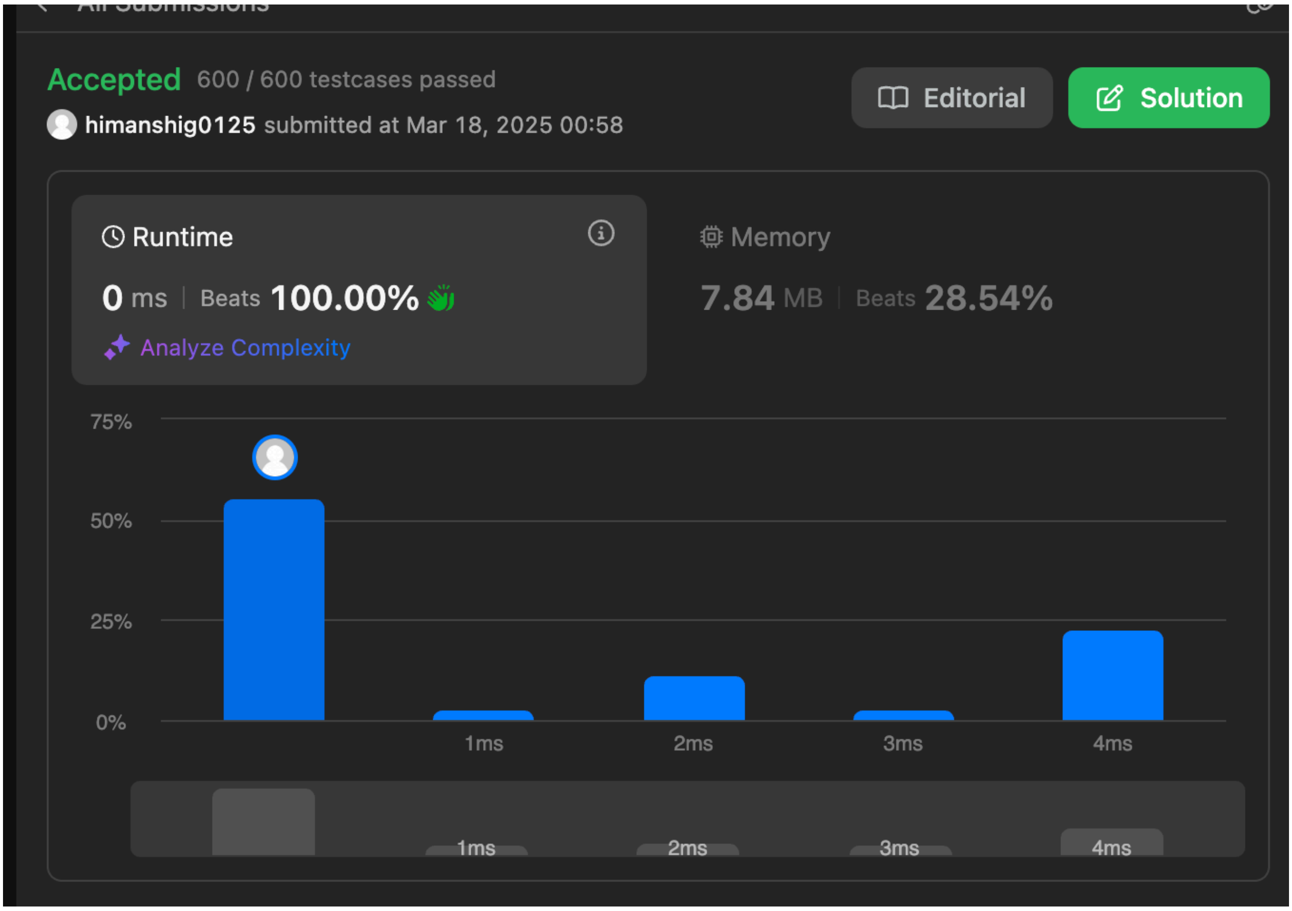
0 ms | Beats 100.00%

Memory

11.59 MB | Beats 80.27%

2. Reverse Bits

```
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;
    }
};
```



Accepted 600 / 600 testcases passed

himanshig0125 submitted at Mar 18, 2025 00:58

Runtime

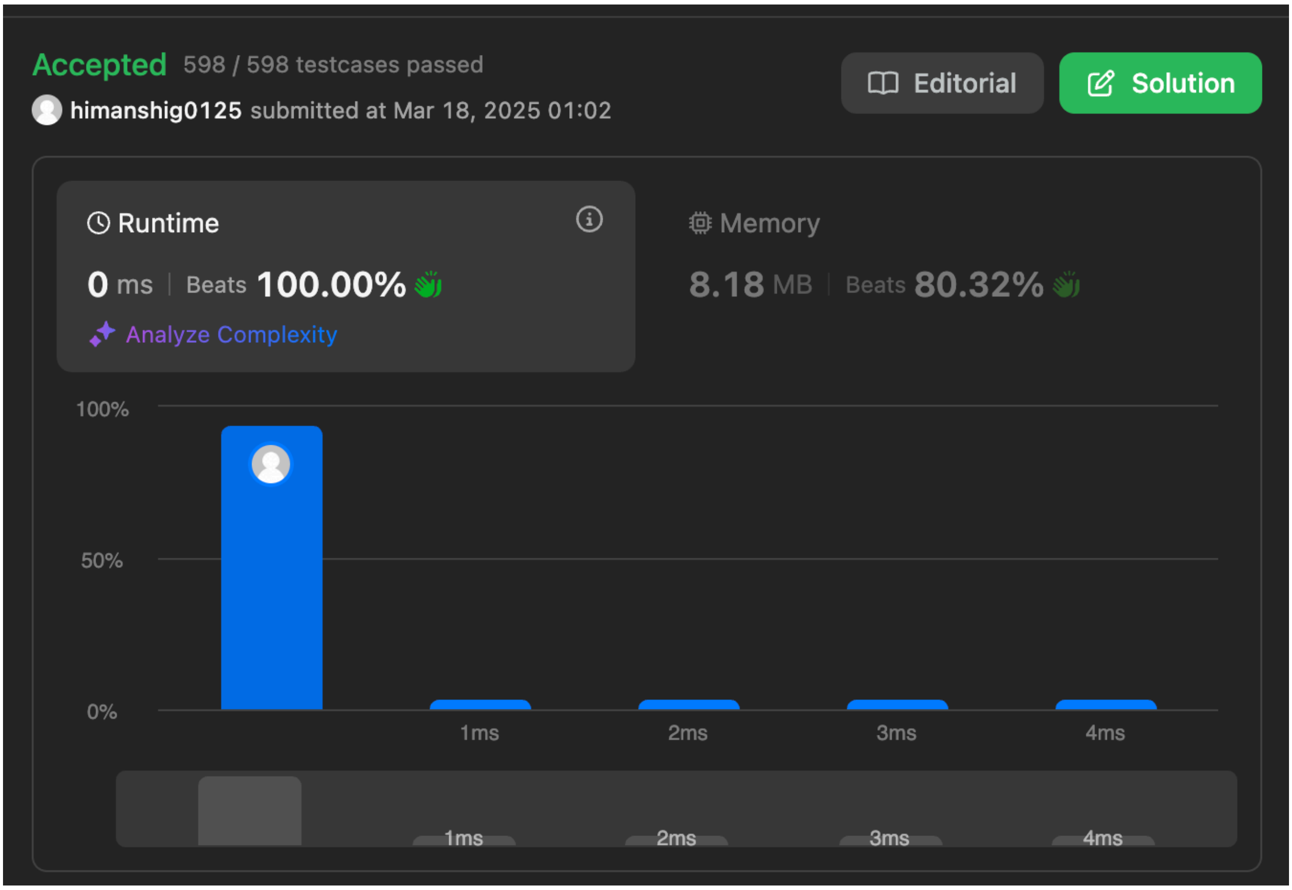
0 ms | Beats 100.00%

Memory

7.84 MB | Beats 28.54%

3. Number of 1 Bits

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



Accepted 598 / 598 testcases passed

himanshig0125 submitted at Mar 18, 2025 01:02

Runtime

0 ms | Beats 100.00%

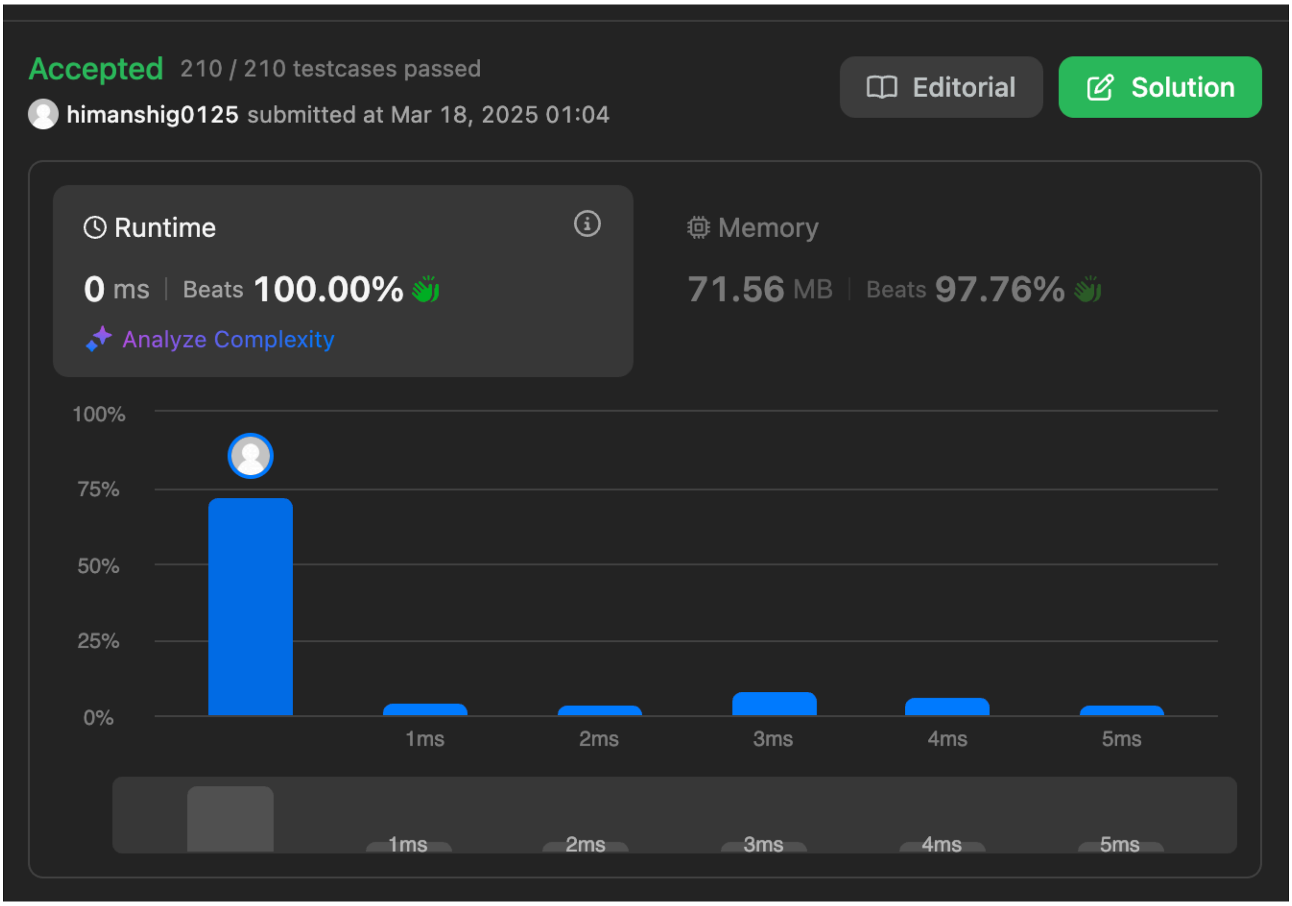
Memory

8.18 MB | Beats 80.32%

4. Maximum Subarray

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int n = nums.size();
        int sum = 0;
        int maxi = INT_MIN;

        for (int i = 0; i < n; i++){
            sum = sum + nums[i];
            maxi = max(maxi, sum);
            if (sum < 0){
                sum = 0;
            }
        }
        return maxi;
    }
};
```



Accepted 210 / 210 testcases passed

himanshig0125 submitted at Mar 18, 2025 01:04

Runtime

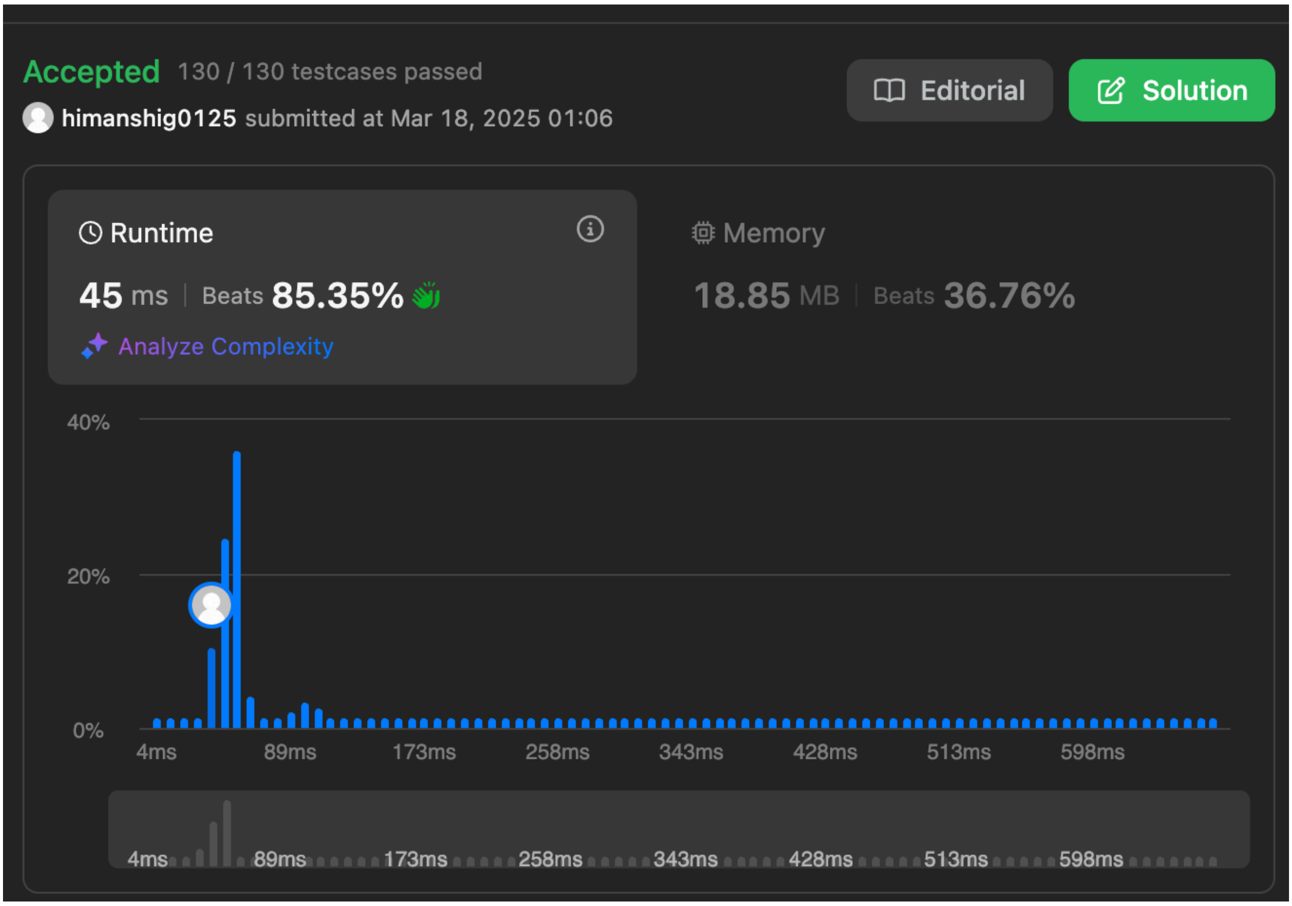
0 ms | Beats 100.00%

Memory

71.56 MB | Beats 97.76%

5. Search a 2D Matrix II

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



Accepted 130 / 130 testcases passed

himanshig0125 submitted at Mar 18, 2025 01:06

Runtime

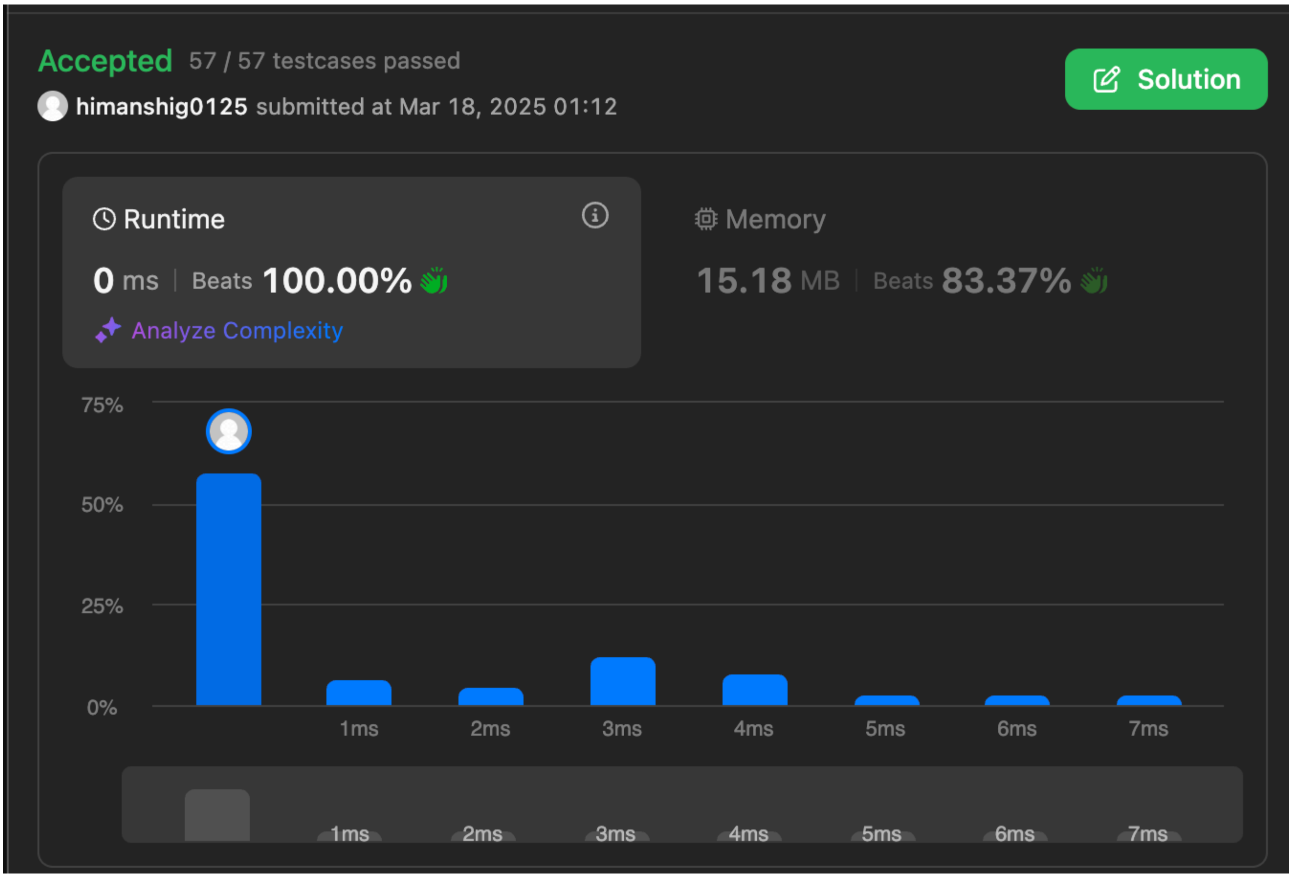
45 ms | Beats 85.35%

Memory

18.85 MB | Beats 36.76%

6. Super Pow

```
class Solution {
public:
    const int MOD = 1337;
    int modPow(int x, int n) {
        int res = 1;
        x %= MOD;
        while (n) {
            if (n % 2) res = (res * x) % MOD;
            x = (x * x) % MOD;
            n /= 2;
        }
        return res;
    }
    int superPow(int a, vector<int>& b) {
        int res = 1;
        for (int digit : b) {
            res = modPow(res, 10) * modPow(a, digit) % MOD;
        }
        return res;
    }
};
```



Accepted 57 / 57 testcases passed

himanshig0125 submitted at Mar 18, 2025 01:12

Runtime

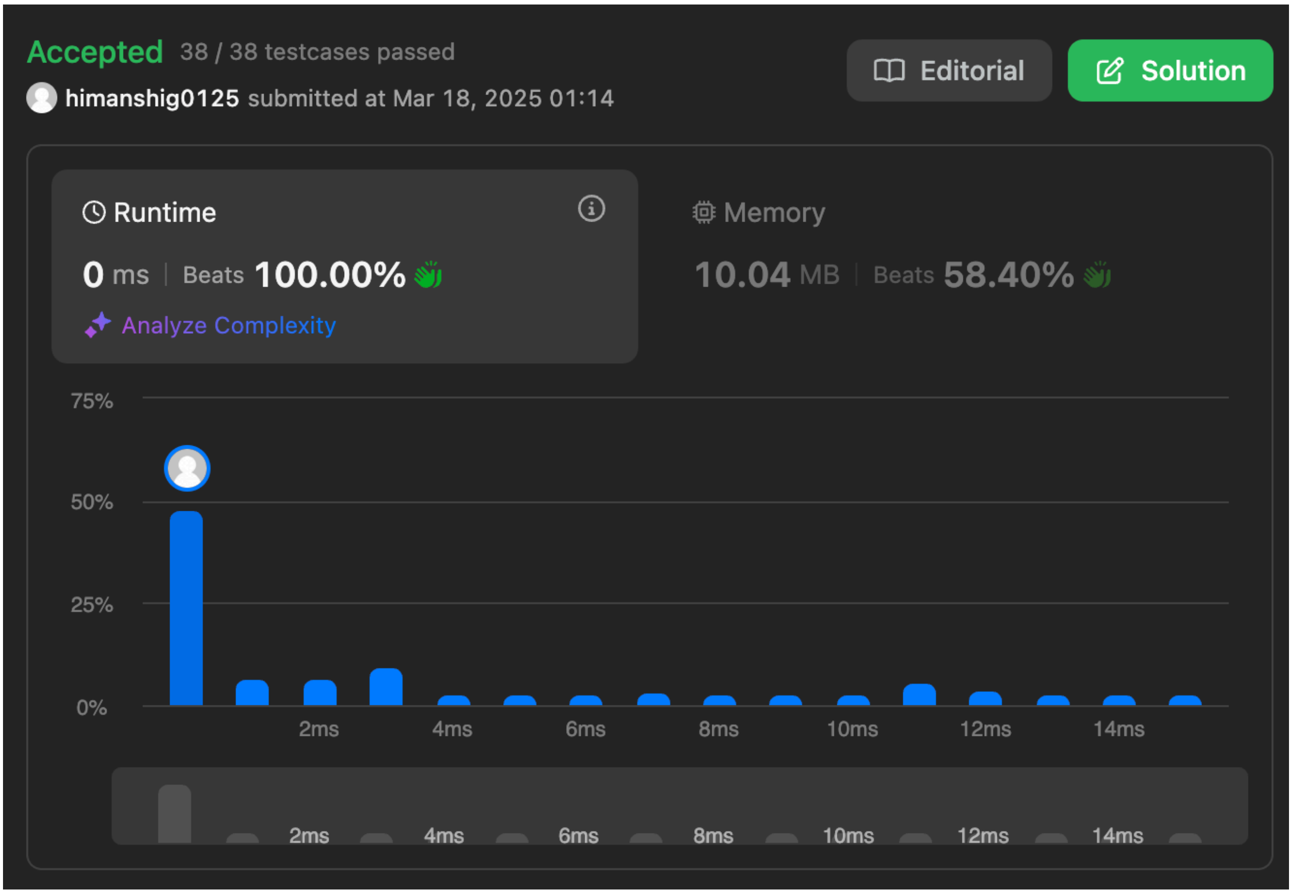
0 ms | Beats 100.00%

Memory

15.18 MB | Beats 83.37%

7. Beautiful Array

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

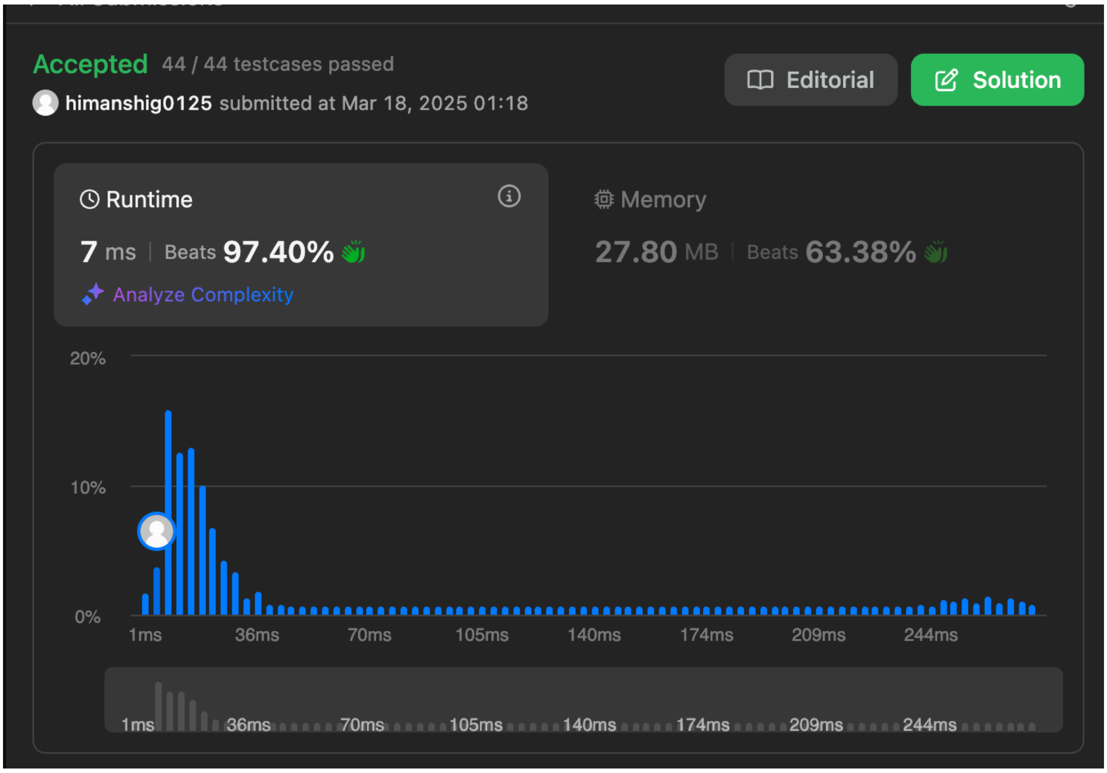


Accepted 38 / 38 testcases passed

himanshig0125 submitted at Mar 18, 2025 01:14

8. The Skyline Problem

```
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<pair<int, int>> events;
        for (auto& b : buildings) {
            events.emplace_back(b[0], -b[2]);
            events.emplace_back(b[1], b[2]);
        }
        sort(events.begin(), events.end());
        multiset<int> heights = {0};
        vector<vector<int>> res;
        int prevMax = 0;
        for (auto& [x, h] : events) {
            if (h < 0) heights.insert(-h);
            else heights.erase(heights.find(h));
            int curMax = *heights.rbegin();
            if (curMax != prevMax) {
                res.push_back({x, curMax});
                prevMax = curMax;
            }
        }
        return res;
    }
};
```



Accepted 44 / 44 testcases passed

himanshig0125 submitted at Mar 18, 2025 01:18

Runtime

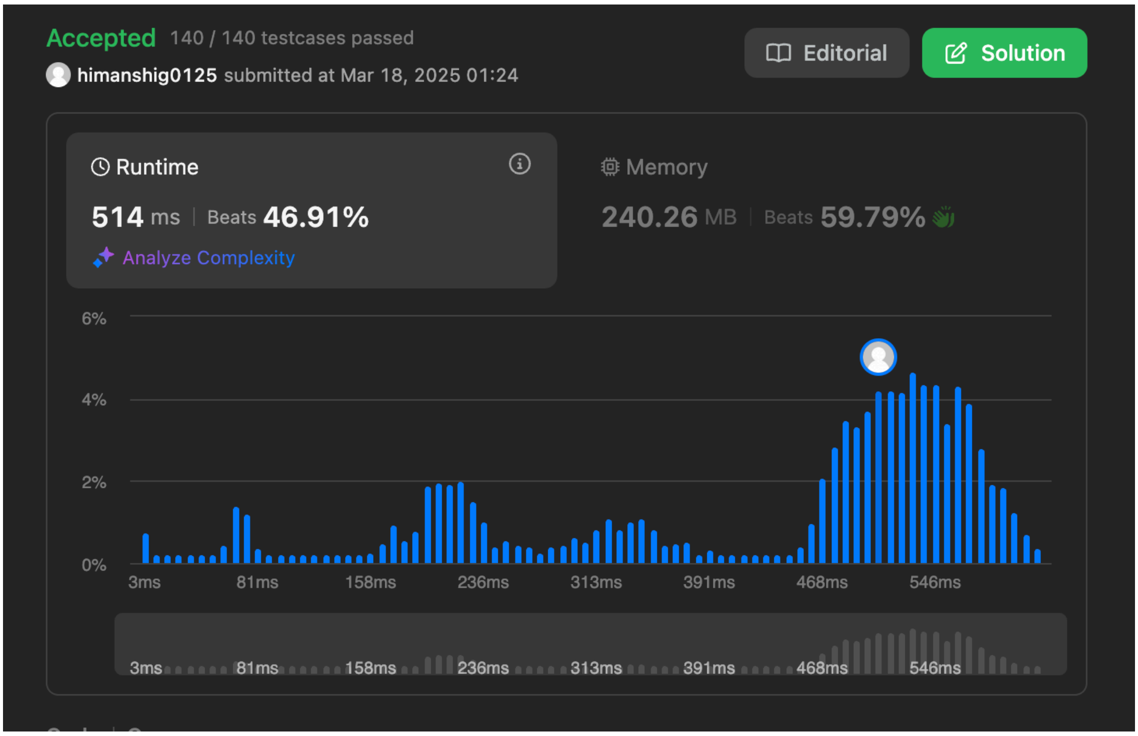
7 ms | Beats 97.40%

Memory

27.80 MB | Beats 63.38%

9. Reverse Pair

```
class Solution {
public:
    int mergeSort(vector<int>& nums, int left, int right) {
        if (left >= right) return 0;
        int mid = left + (right - left) / 2;
        int count = mergeSort(nums, left, mid) + mergeSort(nums, mid + 1, right);
        int j = mid + 1;
        for (int i = left; i <= mid; i++) {
            while (j <= right && nums[i] > 2LL * nums[j]) j++;
            count += (j - (mid + 1));
        }
        vector<int> sorted;
        int i = left, k = mid + 1;
        while (i <= mid && k <= right) {
            if (nums[i] <= nums[k]) sorted.push_back(nums[i++]);
            else sorted.push_back(nums[k++]);
        }
        while (i <= mid) sorted.push_back(nums[i++]);
        while (k <= right) sorted.push_back(nums[k++]);
        for (int i = left; i <= right; i++) nums[i] = sorted[i - left];
        return count;
    }
    int reversePairs(vector<int>& nums) {
        return mergeSort(nums, 0, nums.size() - 1);
    }
};
```



Accepted 140 / 140 testcases passed

himanshig0125 submitted at Mar 18, 2025 01:24

Runtime

514 ms | Beats 46.91%

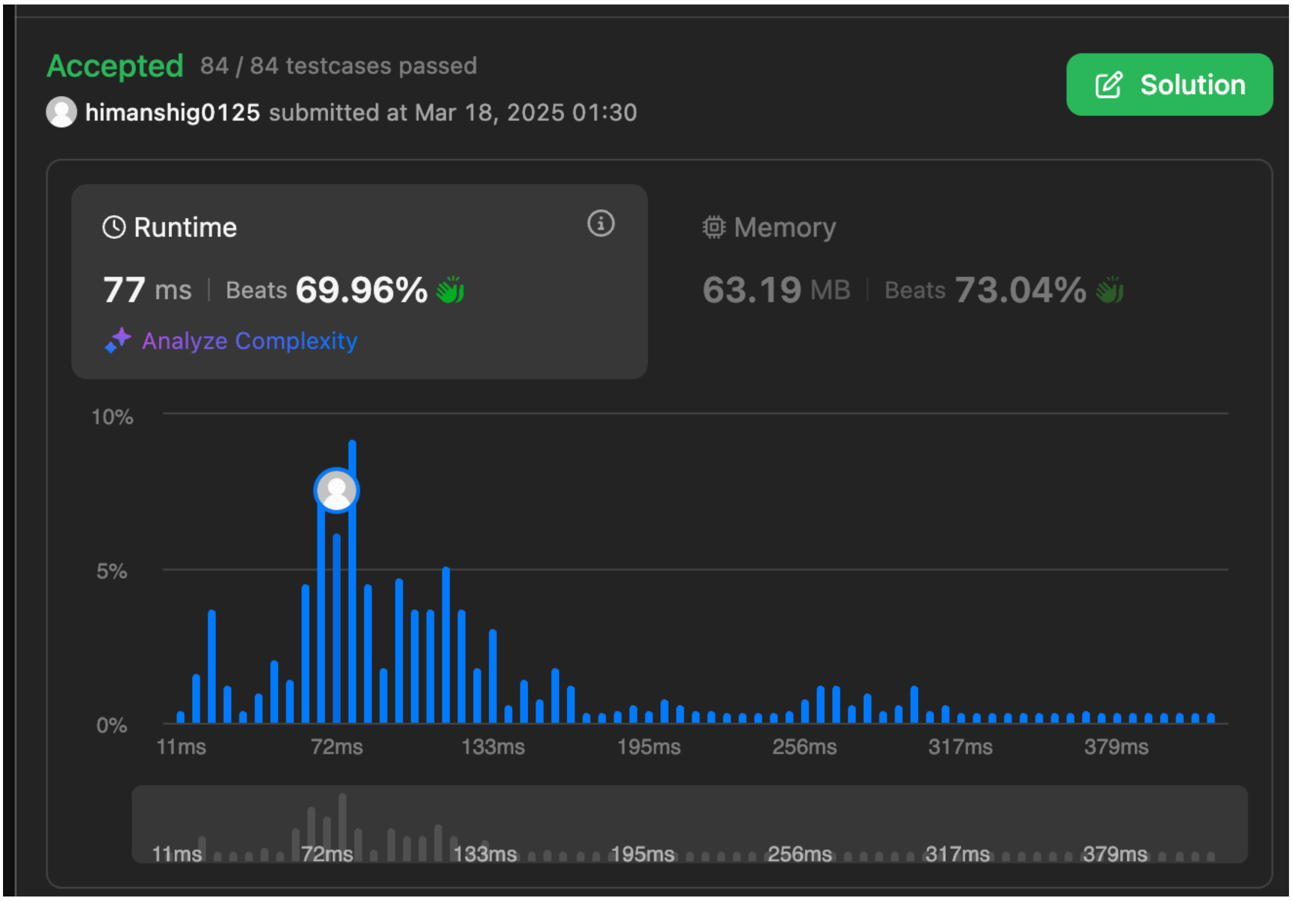
Memory

240.26 MB | Beats 59.79%

10. Longest Increasing Subsequence II

```
class Solution {
public:
    class SegmentTree {
    public:
        vector<int> tree;
        int size;
        SegmentTree(int n) {
            size = n;
            tree.resize(4 * n, 0);
        }
        void update(int index, int value, int node, int start, int end) {
            if (start == end) {
                tree[node] = value;
                return;
            }
            int mid = (start + end) / 2;
            if (index <= mid) update(index, value, 2 * node, start, mid);
            else update(index, value, 2 * node + 1, mid + 1, end);
            tree[node] = max(tree[2 * node], tree[2 * node + 1]);
        }
        int query(int left, int right, int node, int start, int end) {
            if (left > end || right < start) return 0;
            if (left <= start && end <= right) return tree[node];
            int mid = (start + end) / 2;
            return max(query(left, right, 2 * node, start, mid), query(left, right, 2 * node +
1, mid + 1, end));
        }
        void update(int index, int value) {
            update(index, value, 1, 1, size);
        }
        int query(int left, int right) {
            return query(left, right, 1, 1, size);
        }
    };

    int lengthOfLIS(vector<int>& nums, int k) {
        int maxVal = *max_element(nums.begin(), nums.end());
        SegmentTree segTree(maxVal);
        int maxLength = 0;
        for (int num : nums) {
            int bestPrev = segTree.query(max(1, num - k), num - 1);
            int newLength = bestPrev + 1;
            segTree.update(num, newLength);
            maxLength = max(maxLength, newLength);
        }
        return maxLength;
    }
};
```



Accepted 84 / 84 testcases passed

himanshig0125 submitted at Mar 18, 2025 01:30

Runtime

77 ms | Beats 69.96%

Memory

63.19 MB | Beats 73.04%