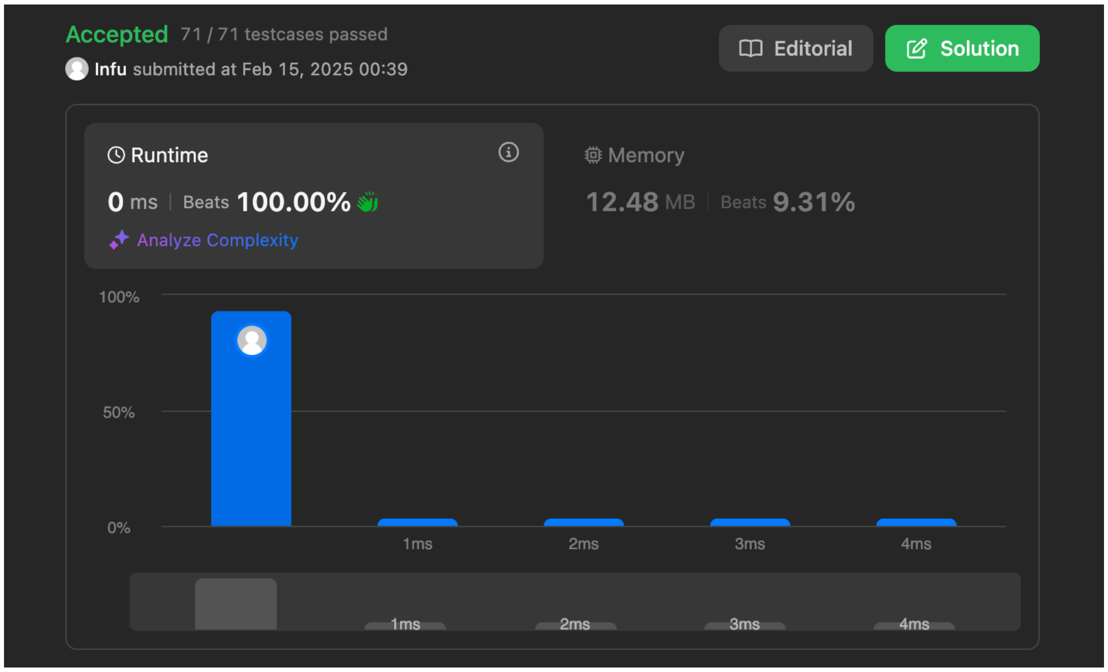
108. Convert Sorted Array to Binary Search Tree

```
class Solution {
    TreeNode* sortedArrayToBST(vector<int>& nums, int start, int end){
        if(end<=start) return NULL;
        int midIdx=(end+start)/2;
        TreeNode* root=new TreeNode(nums[midIdx]);
        root->left=sortedArrayToBST(nums, start, midIdx);
        root->right=sortedArrayToBST(nums, midIdx+1,end);
        return root;
    }
public:
    TreeNode* sortedArrayToBST(vector<int>& nums) {
        return sortedArrayToBST(nums, 0,nums.size());
    }
};
```



Accepted 71 / 71 testcases passed

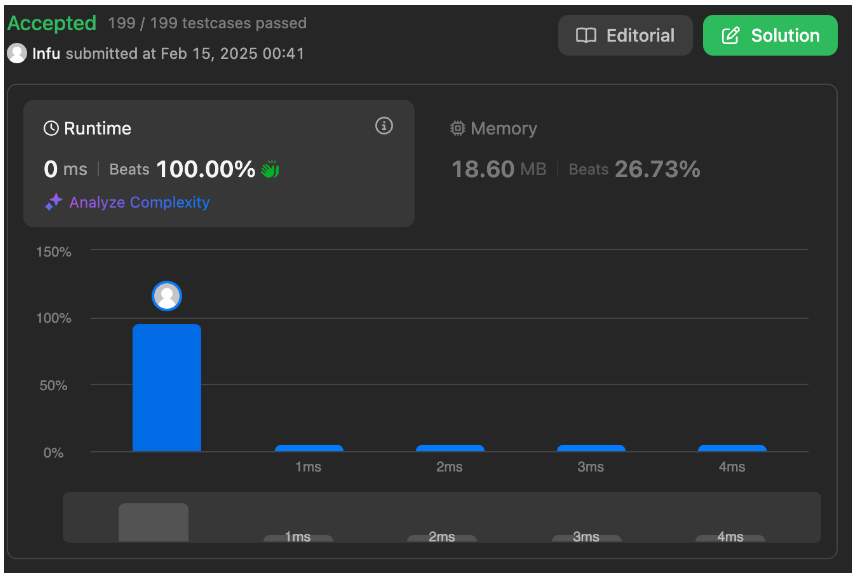
Infu submitted at Feb 15, 2025 00:39

0 ms | Beats 100.00%

12.48 MB | Beats 9.31%

191. Number of 1 Bits

```
class Solution {
public:
    int hammingWeight(int n) {
        stack<int> s;
        while(n){
            s.push(n % 2);
            n = n / 2;
        }
        int count = 0;
        while(!s.empty()){
            if(s.top() == 1) count++;
            s.pop();
        }
        return count;
    }
};
```


Accepted 199 / 199 testcases passed
Infu submitted at Feb 15, 2025 00:41

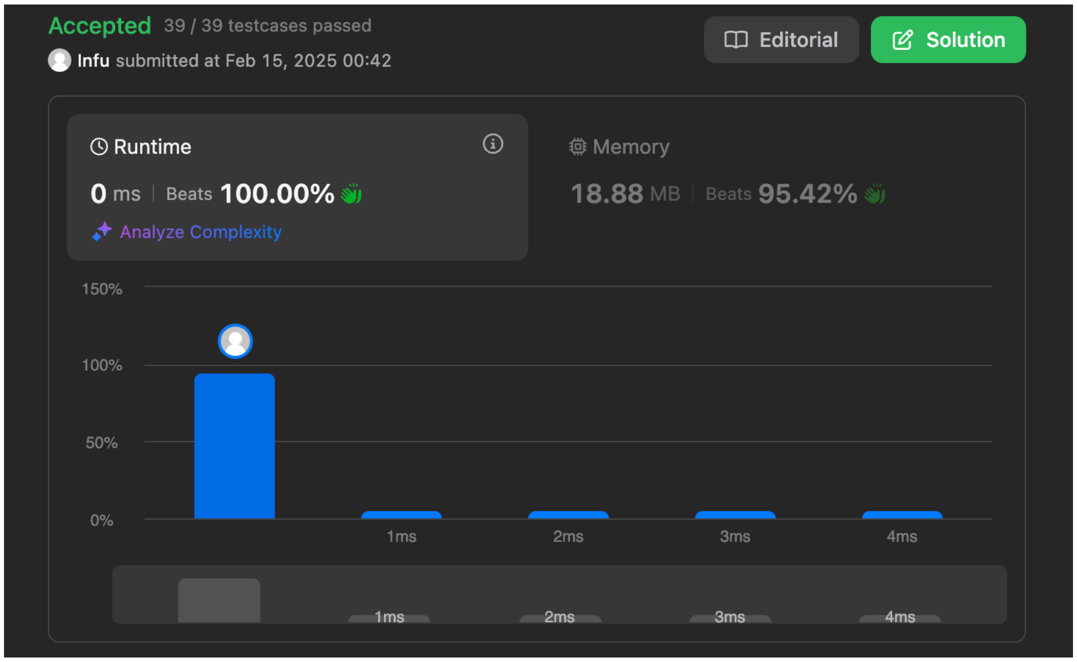
Runtime
0 ms | Beats 100.00%

Memory
18.60 MB | Beats 26.73%

912. Sort an Array

```
class Solution {
public:
    void outPlaceMerge(vector<int> &nums, int low, int mid, int high)
{
        if (low >= high) return;
        int l = low, r = mid + 1, k = 0, size = high - low + 1;
        vector<int> sorted(size, 0);
        while (l <= mid and r <= high)
            sorted[k++] = nums[l] < nums[r] ? nums[l++] : nums[r++];
        while (l <= mid)
            sorted[k++] = nums[l++];
        while (r <= high)
            sorted[k++] = nums[r++];
        for (k = 0; k < size; k++)
            nums[k + low] = sorted[k];
    }

    void mergeSort(vector<int> &nums, int low, int high) {
        if (low >= high) return;
        int mid = (high - low) / 2 + low;
        mergeSort(nums, low, mid);
        mergeSort(nums, mid + 1, high);
        outPlaceMerge(nums, low, mid, high);
    }
    vector<int> sortArray(vector<int>& nums) {
        mergeSort(nums, 0, nums.size() - 1);
        return nums;
    }
};
```


Accepted 39 / 39 testcases passed

Infu submitted at Feb 15, 2025 00:42

Runtime

0 ms | Beats 100.00%

Memory

18.88 MB | Beats 95.42%

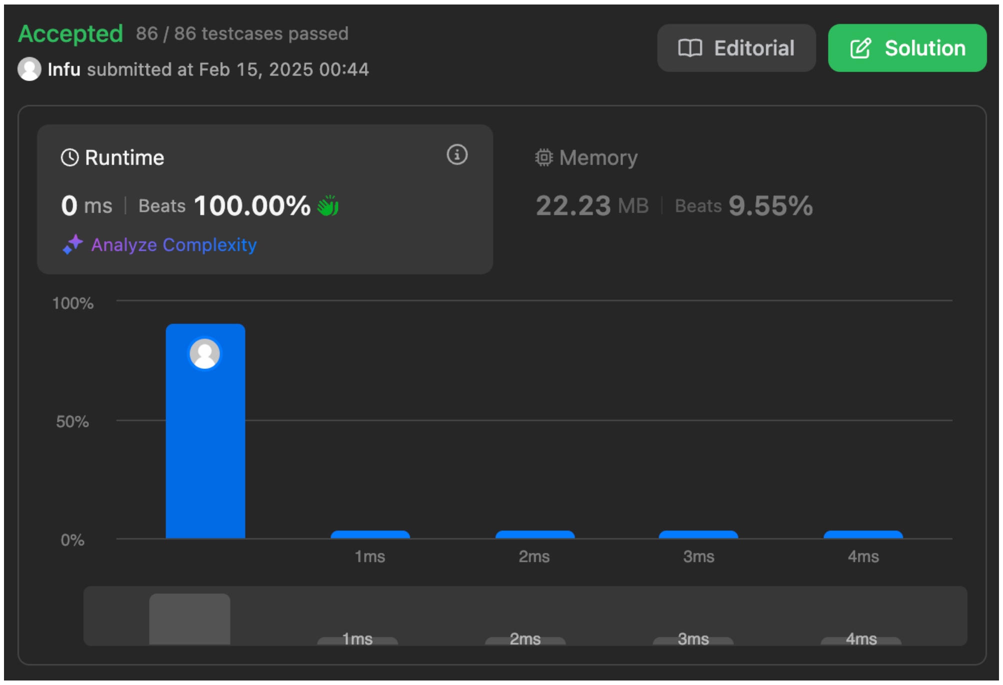
53. Maximum Subarray

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

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

        sum += arr[i];
        if (sum > maxi) {
            maxi = sum;
        }
        if (sum < 0) {
            sum = 0;
        }
    }

    return maxi;
    }
};
```


Accepted 86 / 86 testcases passed
Infu submitted at Feb 15, 2025 00:44

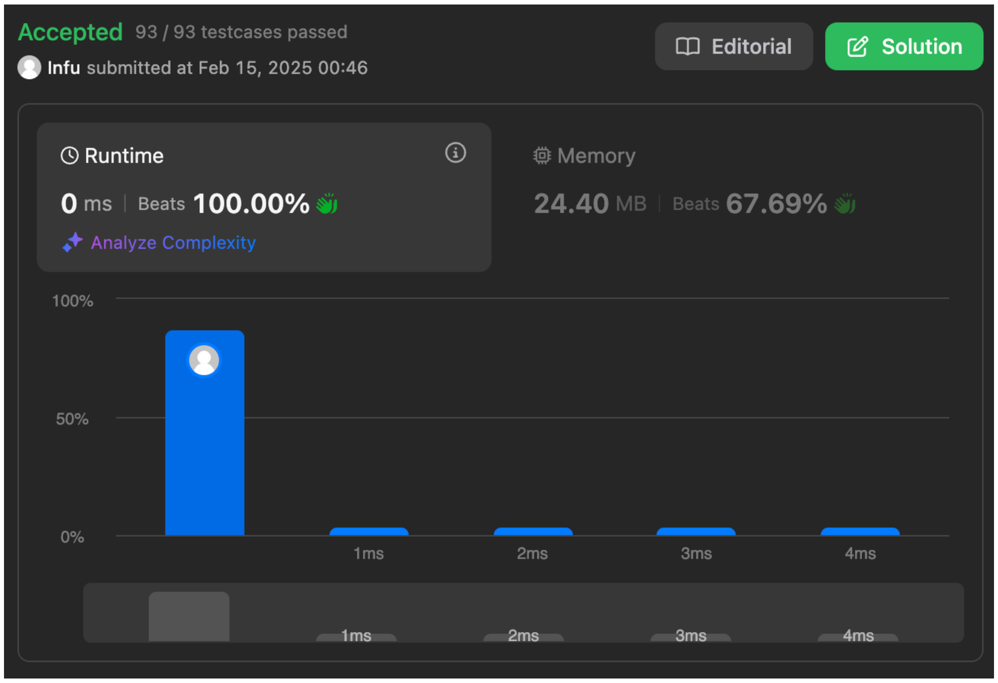
932. Beautiful Array

```
class Solution {
public:
    vector<int> beautifulArray(int n) {
        if (n==1) return {1};
        vector<int> arr = beautifulArray(n-1);

        vector<int> res;
        for (auto i: arr)
            if (2*i - 1 <= n)
                res.push_back(2*i-1);

        for (auto i: arr)
            if (2*i <= n)
                res.push_back(2*i);

        return res;
    }
};
```



Accepted 93 / 93 testcases passed

Infu submitted at Feb 15, 2025 00:46

Runtime

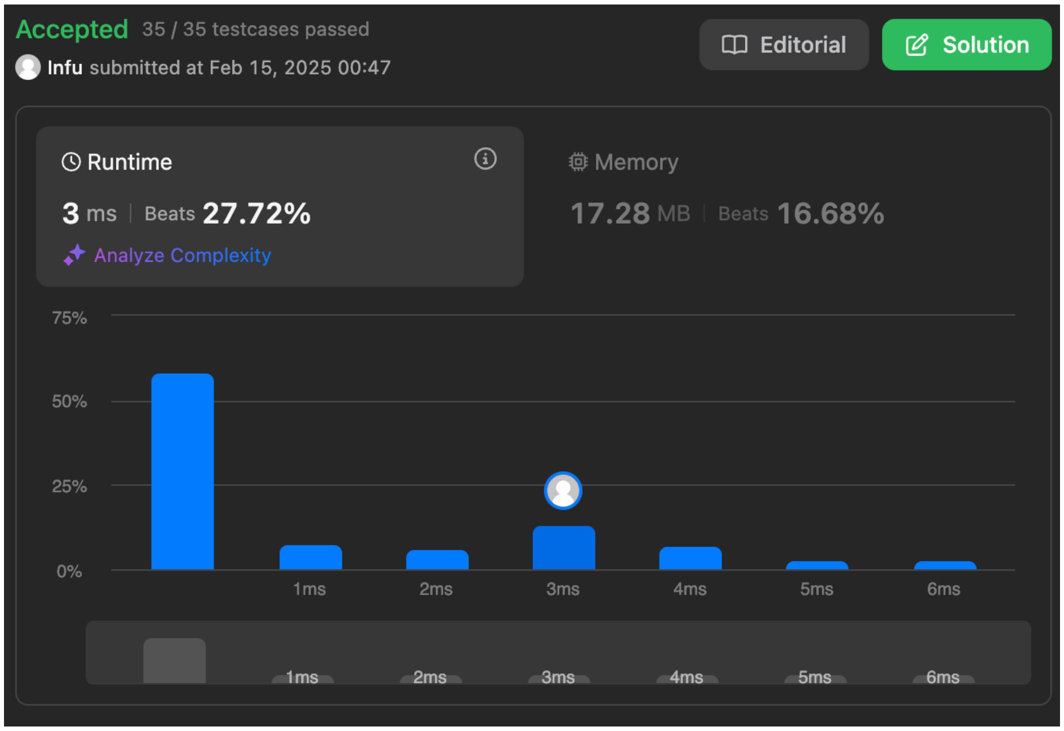
0 ms | Beats 100.00%

Memory

24.40 MB | Beats 67.69%

372. Super Pow

```
class Solution {
public:
    int pow(int a, int b){
        if(b==0) return 1;
        int temp=pow(a,b/2);
        if(b%2==0) return ((temp%1337)*temp%1337)%1337;
        else return (a%1337*((temp%1337*temp%1337)%1337))%1337;
    }
    int superPow(int a, vector<int>& b) {
        if(b.size()==0) return 1;
        int x=b.back(); b.pop_back();
        return pow(superPow(a, b), 10) * pow(a, x) % 1337;
    }
};
```


Accepted 35 / 35 testcases passed

Infu submitted at Feb 15, 2025 00:47

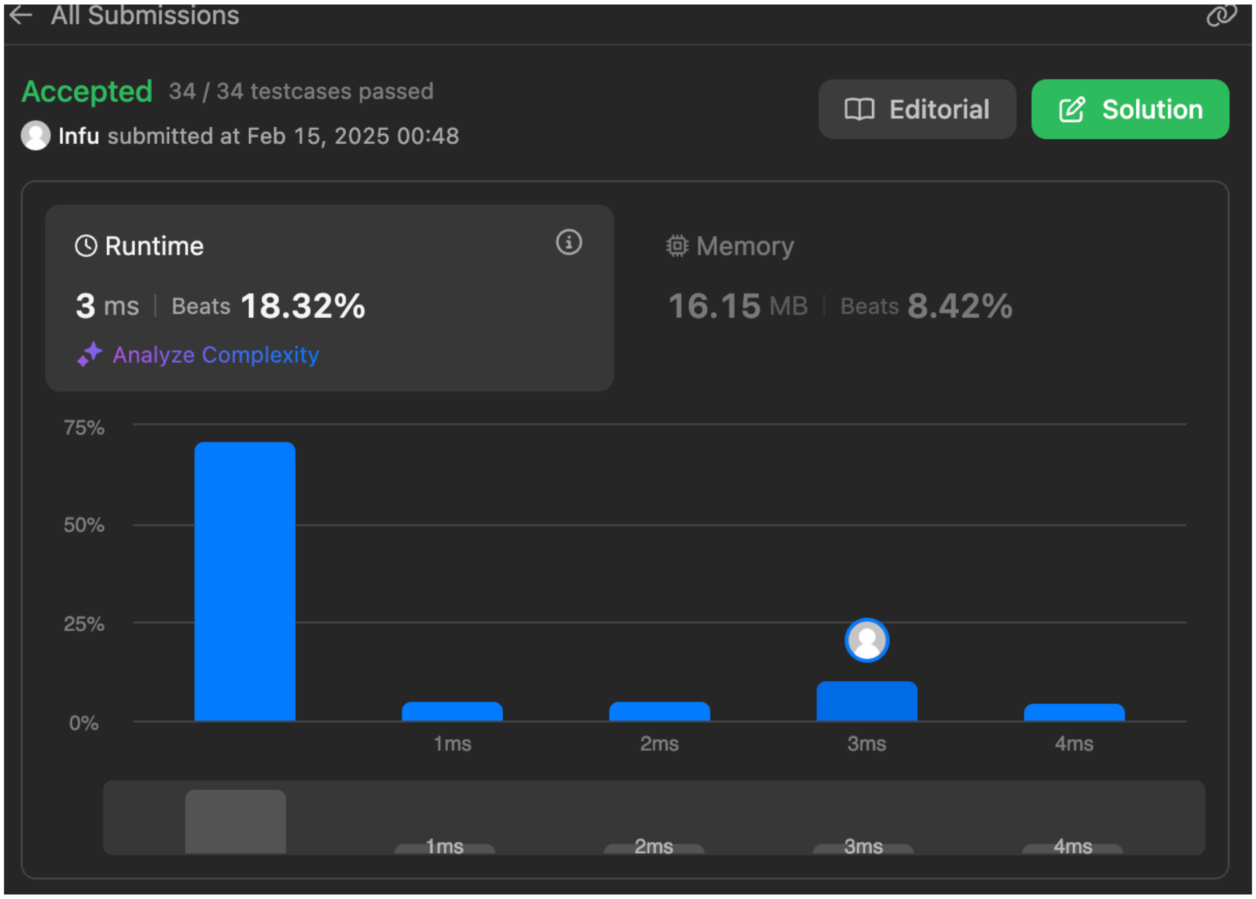
Runtime
3 ms | Beats 27.72%

Memory
17.28 MB | Beats 16.68%

218. The Skyline Problem

```
class Solution {
public:
     vector<vector<int>> getSkyline(vector<vector<int>>& b) {
        priority_queue<vector<int>> live;
        int n=b.size();

        int cur=0;
        vector<vector<int>> ans;
        while(cur<n || !live.empty()){
            int cur_x=live.empty()?b[cur][0]:live.top()[1];
            if(cur>=n || b[cur][0]>cur_x){
                while(!live.empty() && (live.top()[1]<=cur_x)){
                    live.pop();
                }
            }
            else{
                cur_x=b[cur][0];
                while(cur<n && cur_x==b[cur][0]){
                    live.push({b[cur][2],b[cur][1]});
                    cur++;
                }
            }
            int cur_h=live.empty()?0:live.top()[0];
            if(ans.empty() || ans[ans.size()-1][1]!=cur_h){
                ans.push_back({cur_x,cur_h});
            }
        }
        return ans;
    }
};
```



All Submissions

Accepted 34 / 34 testcases passed

Infu submitted at Feb 15, 2025 00:48

Runtime

3 ms | Beats 18.32%

Memory

16.15 MB | Beats 8.42%