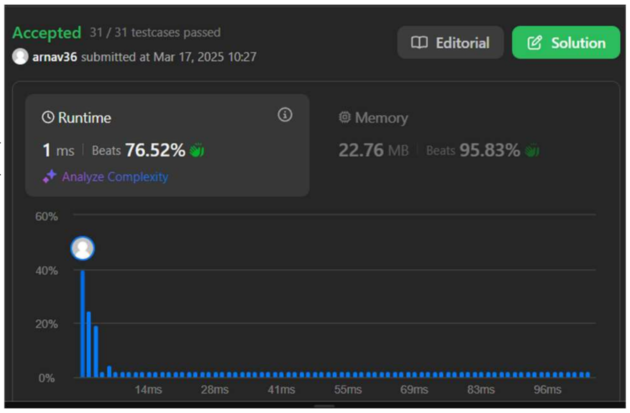
ASSIGNMENT 6

STUDENT NAME: Arnav Jain **UID:** 22BCS15161
BRANCH: CSE **SECTION:** 22BCS_FL_IOT_601A
SEMESTER: 6 **DATE OF SUBMISSION:** 4/3/25
SUBJECT NAME: AP LAB -2 **SUBJECT CODE:** 22CSP-351

LEET CODE QUESTIONS :

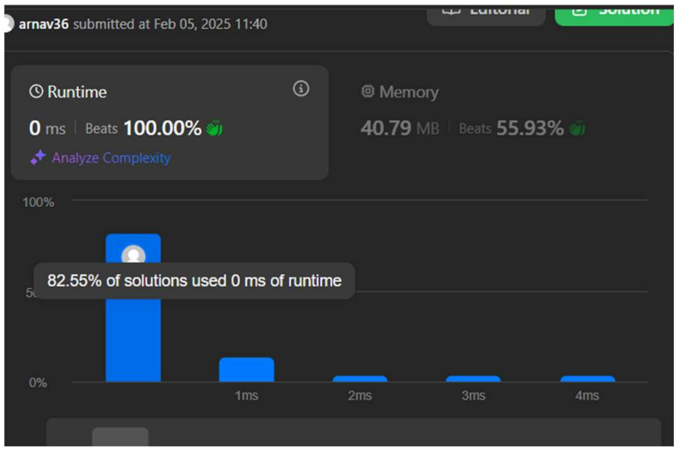
108.Convert Sorted Array to Binary Search Tree

```
class Solution {
public:
      TreeNode* bt(vector<int>& nums, int s, int e) {
        if (s > e) return nullptr;
        int n = (s + e) / 2;
        TreeNode* root = new TreeNode(nums[n]);

        root->left = bt(nums, s, n - 1);
        root->right = bt(nums, n + 1, e);

        return root;
    }

    TreeNode* sortedArrayToBST(vector<int>& nums) {
        return bt(nums, 0, nums.size() - 1);
    }
};
```


Accepted 31 / 31 testcases passed
arnav36 submitted at Mar 17, 2025 10:27
Runtime
1 ms | Beats 76.52%
Memory
22.76 MB | Beats 95.83%

191. Number of 1 Bits

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


arnav36 submitted at Feb 05, 2025 11:40

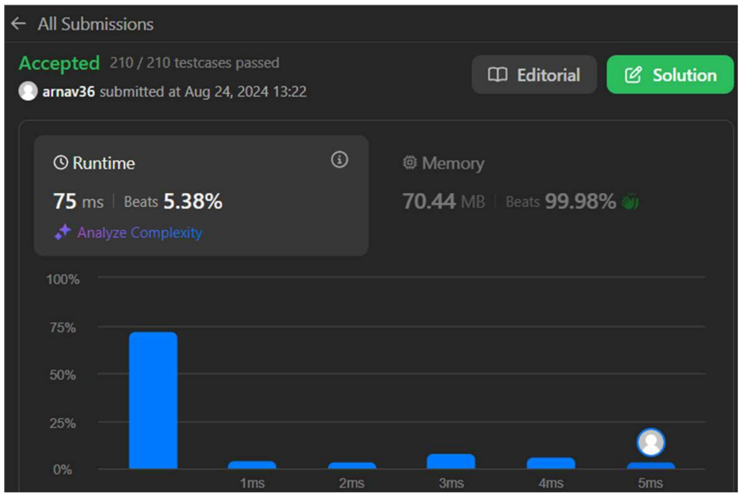
Runtime
0 ms | Beats 100.00%

Memory
40.79 MB | Beats 55.93%

82.55% of solutions used 0 ms of runtime

53. Maximum Subarray

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

        }
        return msum;
    }
};
```


All Submissions

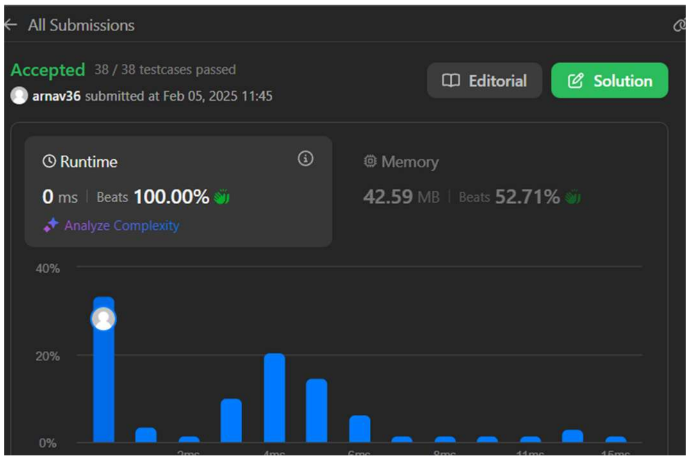
Accepted 210 / 210 testcases passed
arnav36 submitted at Aug 24, 2024 13:22

Runtime
75 ms | Beats 5.38%

Memory
70.44 MB | Beats 99.98%

932.[Beautiful Array]

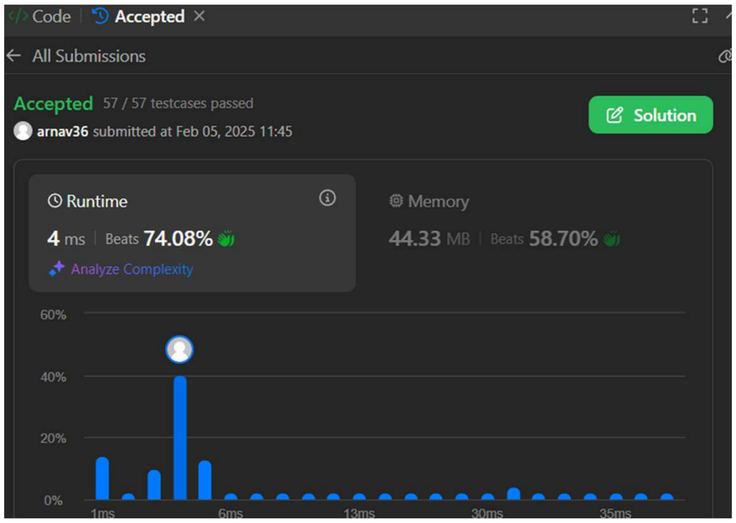
```
class Solution {
    public int[] beautifulArray(int n) {
        if (n == 1) return new int[]{1};

        int[] left = beautifulArray((n + 1) / 2);
        int[] right = beautifulArray(n / 2);

        int[] result = new int[n];
        int index = 0;

        for (int x : left) result[index++] = 2 * x - 1;
        for (int x : right) result[index++] = 2 * x;

        return result;
    }
}
```


All Submissions

Accepted 38 / 38 testcases passed
arnav36 submitted at Feb 05, 2025 11:45

Runtime
0 ms | Beats 100.00%

Memory
42.59 MB | Beats 52.71%


Code | Accepted

All Submissions

Accepted 57 / 57 testcases passed
arnav36 submitted at Feb 05, 2025 11:45

Runtime
4 ms | Beats 74.08%

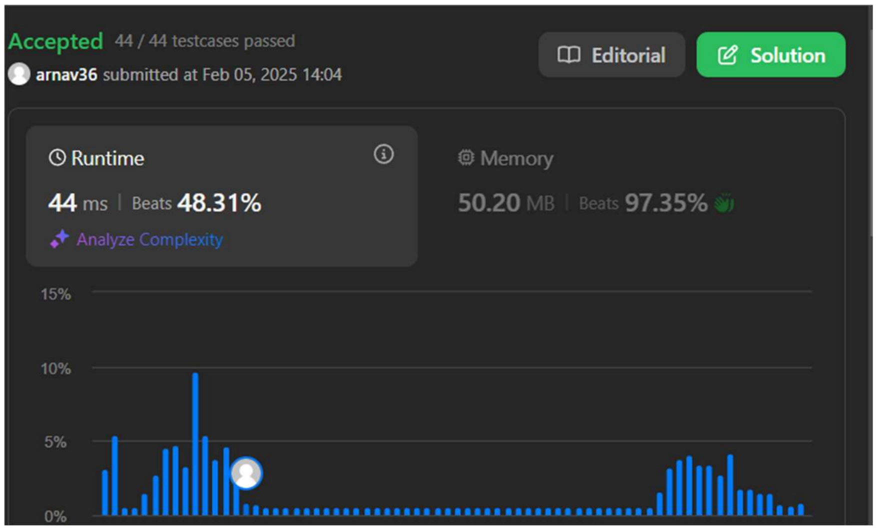
Memory
44.33 MB | Beats 58.70%

372.Super Pow

```
class Solution {
    private static final int MOD = 1337;

    public int superPow(int a, int[] b) {
        int result = 1;
        a %= MOD;

        for (int digit : b) {
            result = (pow(result, 10) * pow(a, digit)) % MOD;
        }

        return result;
    }

    private int pow(int x, int n) {
        int result = 1;

        while (n > 0) {
            if ((n & 1) == 1) result = (result * x) % MOD;
            x = (x * x) % MOD;
            n >>= 1;
        }

        return result;
    }
}
```

218.The Skyline Problem

```
class Solution {
    public List<List<Integer>> getSkyline(int[][] buildings) {
        List<int[]> heights = new ArrayList<>();
        for (int[] b : buildings) {
            heights.add(new int[]{b[0], -b[2]});
            heights.add(new int[]{b[1], b[2]});
        }

        Collections.sort(heights, (a, b) -> a[0] != b[0] ? a[0] - b[0] : a[1] - b[1]);
        TreeMap<Integer, Integer> heightMap = new TreeMap<>(Collections.reverseOrder());
        heightMap.put(0, 1);

        List<List<Integer>> result = new ArrayList<>();
        int prevMaxHeight = 0;

        for (int[] h : heights) {
            if (h[1] < 0) {
                heightMap.put(-h[1], heightMap.getOrDefault(-h[1], 0) + 1);
            } else {
                heightMap.put(h[1], heightMap.get(h[1]) - 1);
                if (heightMap.get(h[1]) == 0) heightMap.remove(h[1]);
            }

            int currMaxHeight = heightMap.firstKey();
            if (currMaxHeight != prevMaxHeight) {
                result.add(Arrays.asList(h[0], currMaxHeight));
                prevMaxHeight = currMaxHeight;
            }
        }

        return result;
    }
}
```


Accepted 44 / 44 testcases passed
arnav36 submitted at Feb 05, 2025 14:04

Runtime
44 ms | Beats 48.31%

Memory
50.20 MB | Beats 97.35%