

Name: Archishman Mukherjee uid: 22bcs11671 sec-605(B)

108. Convert Sorted Array to Binary Search Tree

The screenshot shows a C++ solution for the problem "Convert Sorted Array to Binary Search Tree". The code is in a dark-themed editor. The left sidebar shows the problem status as "Accepted" with 31/31 testcases passed. The runtime is 1 ms, beating 76.74% of solutions, and memory is 22.84 MB, beating 81.23%. A bar chart shows the runtime distribution. The code defines a `TreeNode` struct and a recursive function `sortedArrayToBST` that converts a sorted array into a balanced BST. The test result shows two test cases: `[-10,-3,0,5,9]` and `[1,3]`.

```
Accepted 31 / 31 testcases passed
Temporary46 submitted at Mar 22, 2025 19:25

Runtime 1 ms | Beats 76.74%
Memory 22.84 MB | Beats 81.23%

Code C++

/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 * };
 */

TreeNode* sortedArrayToBST(vector<int>& nums, int start, int end) {
    if(start > end) {
        return NULL;
    }
    int mid = (start + end) / 2;
    TreeNode* root = new TreeNode(nums[mid]);

    root->left = sortedArrayToBST(nums, start, mid - 1);
    root->right = sortedArrayToBST(nums, mid + 1, end);

    return root;
};
```

Testcase 1: `[-10,-3,0,5,9]`
Testcase 2: `[1,3]`

191. Number of 1 Bits

The screenshot shows a C++ solution for the problem "Number of 1 Bits". The code is in a dark-themed editor. The left sidebar shows the problem status as "Accepted" with 598/598 testcases passed. The runtime is 0 ms, beating 100.00% of solutions, and memory is 8.36 MB, beating 12.15%. A bar chart shows the runtime distribution. The code defines a `Solution` class with a `hammingWeight` method that counts the number of 1 bits in a 32-bit integer using a loop. The test result shows three test cases: `11`, `128`, and `2147483645`.

```
Accepted 598 / 598 testcases passed
Temporary46 submitted at Mar 22, 2025 19:26

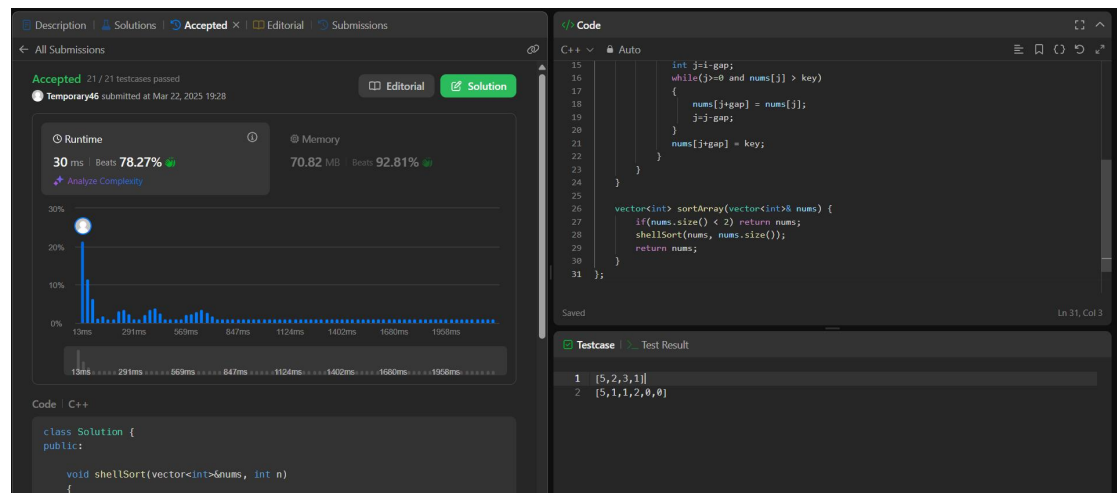
Runtime 0 ms | Beats 100.00%
Memory 8.36 MB | Beats 12.15%

Code C++

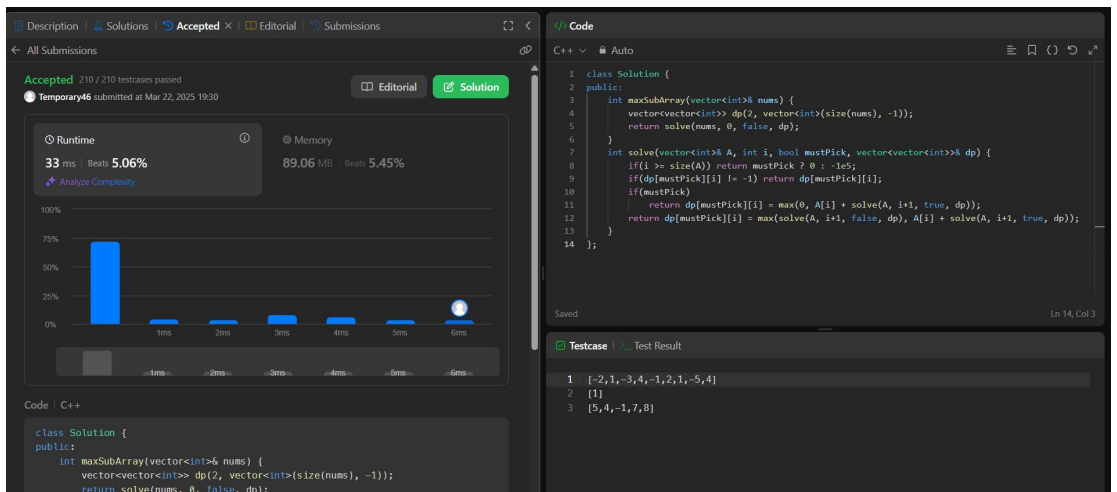
class Solution {
public:
    int hammingWeight(uint32_t n) {
        int res = 0;
        for (int i = 0; i < 32; i++) {
            if ((n >> i) & 1) {
                res += 1;
            }
        }
        return res;
    }
};
```

Testcase 1: `11`
Testcase 2: `128`
Testcase 3: `2147483645`

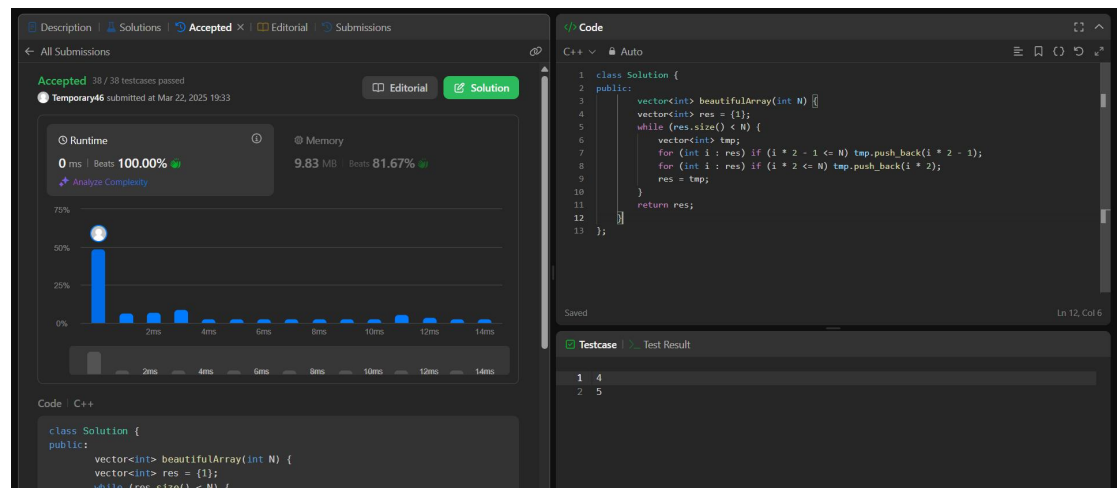
912. Sort an Array



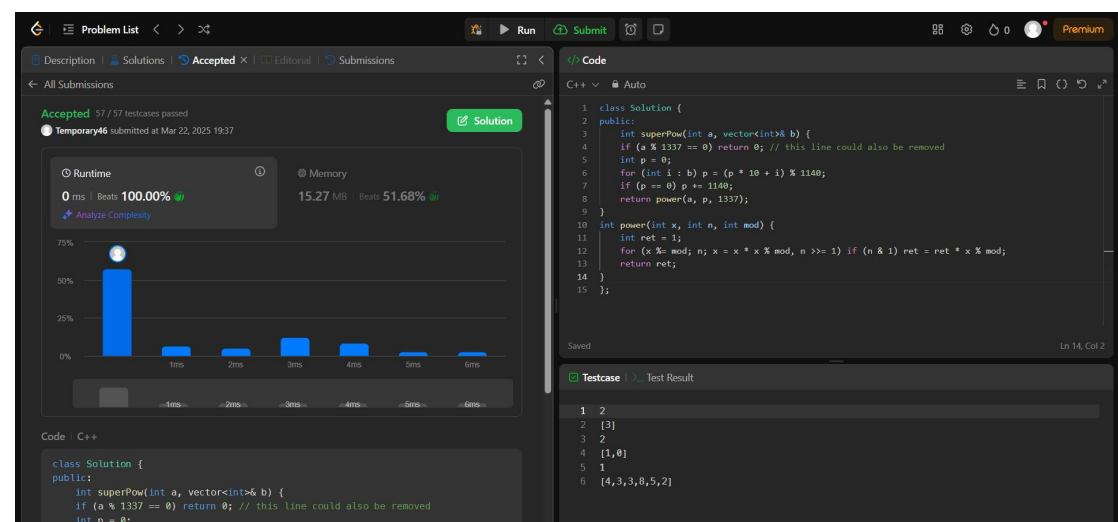
53. Maximum Subarray



932. Beautiful Array



372. Super Pow



218. The Skyline Problem

