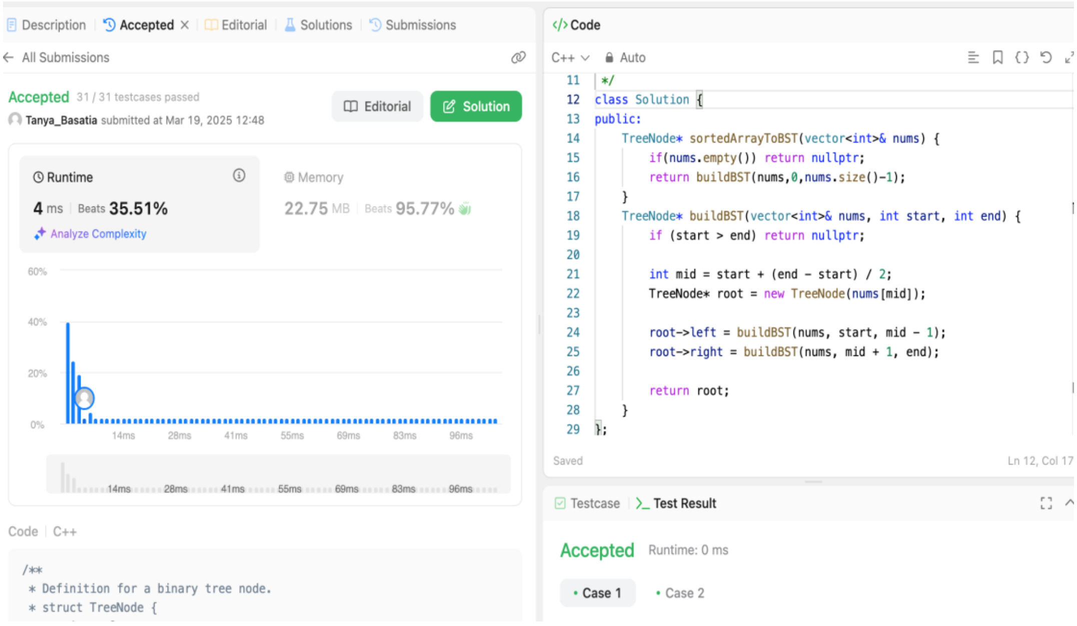
ADVANCED PROGRAMMING LAB ASSIGNMENT-6

Name-Tanya UID-22BCS15446

1. Convert Sorted Array to Binary Search Tree

```
class Solution {
public:
TreeNode* sortedArrayToBST(vector<int>& nums) {
if(nums.empty()) return nullptr;
return buildBST(nums,0,nums.size()-1);
}
TreeNode* buildBST(vector<int>& nums, int start, int end) {
if (start > end) return nullptr;

int mid = start + (end - start) / 2;
TreeNode* root = new TreeNode(nums[mid]);

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

return root;
}
};
```



Accepted 31 / 31 testcases passed

Tanya_Basatia submitted at Mar 19, 2025 12:48

4 ms | Beats 35.51%

22.75 MB | Beats 95.77%

Code | C++

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
```

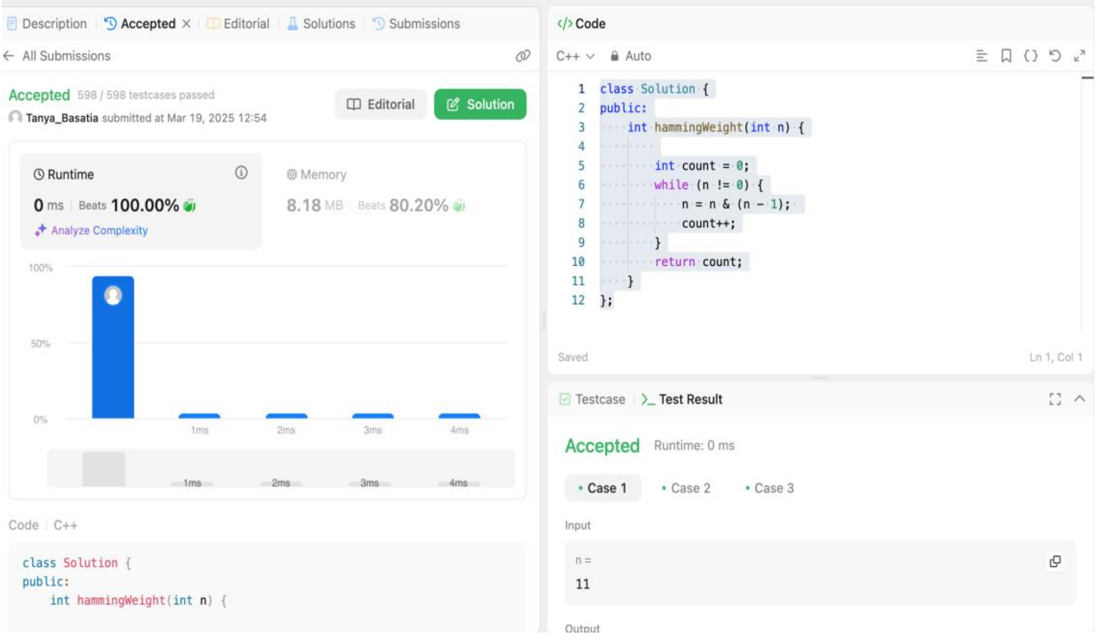
```
 */
class Solution {
public:
    TreeNode* sortedArrayToBST(vector<int>& nums) {
        if(nums.empty()) return nullptr;
        return buildBST(nums,0,nums.size()-1);
    }
    TreeNode* buildBST(vector<int>& nums, int start, int end) {
        if (start > end) return nullptr;

        int mid = start + (end - start) / 2;
        TreeNode* root = new TreeNode(nums[mid]);

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

        return root;
    }
};
```

Saved Ln 12, Col 17

Accepted Runtime: 0 ms

2. Number of 1 Bits

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



Accepted 598 / 598 testcases passed

Tanya_Basatia submitted at Mar 19, 2025 12:54

Runtime 0 ms | Beats 100.00%

Memory 8.18 MB | Beats 80.20%

Code | C++

```
class Solution {
public:
    int hammingWeight(int n) {
```

```
class Solution {
public:
    int hammingWeight(int n) {

        int count = 0;
        while (n != 0) {
            n = n & (n - 1);
            count++;
        }
        return count;
    }
};
```

Accepted Runtime: 0 ms

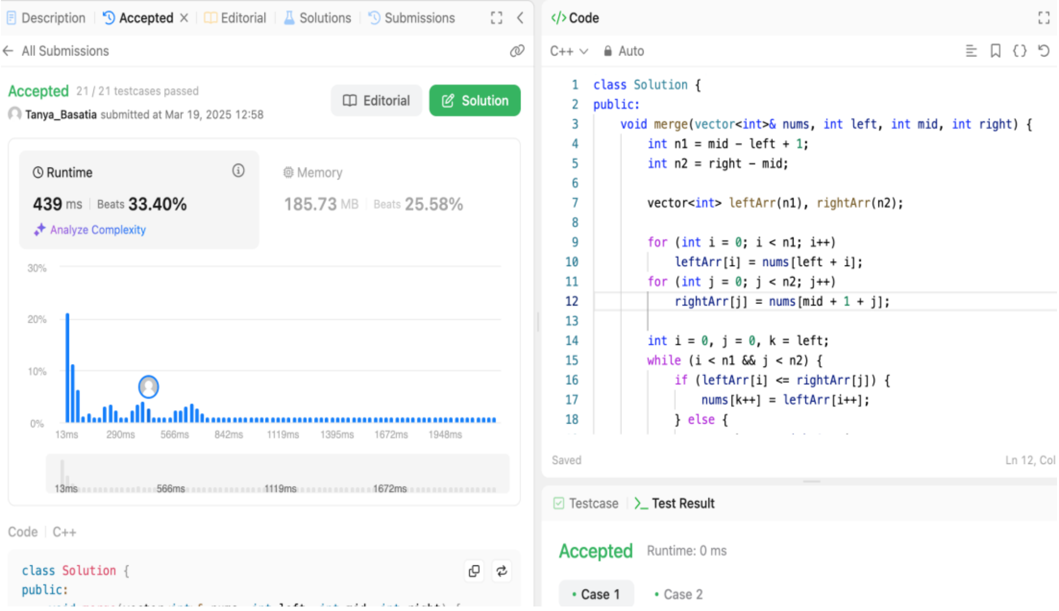
Input

n =
11

3. Sort an array

```
class Solution {
public:
void merge(vector<int>& nums, int left, int mid, int right) {
int n1 = mid - left + 1;
int n2 = right - mid;
vector<int> leftArr(n1), rightArr(n2);
for (int i = 0; i < n1; i++)
leftArr[i] = nums[left + i];
for (int j = 0; j < n2; j++)
rightArr[j] = nums[mid + 1 + j];
int i = 0, j = 0, k = left;
while (i < n1 && j < n2) {
if (leftArr[i] <= rightArr[j]) {
nums[k++] = leftArr[i++];
} else {
```

```
nums[k++] = rightArr[j++];
}
}
while (i < n1) {
nums[k++] = leftArr[i++];
}
while (j < n2) {
nums[k++] = rightArr[j++];
}
}
void mergeSort(vector<int>& nums, int left, int right) {
if (left >= right)
return;
int mid = left + (right - left) / 2;
mergeSort(nums, left, mid);
mergeSort(nums, mid + 1, right);
merge(nums, left, mid, right);
}
vector<int> sortArray(vector<int>& nums) {
mergeSort(nums, 0, nums.size() - 1);
return nums;
}
};
```



Accepted 21 / 21 testcases passed

Tanya_Basatia submitted at Mar 19, 2025 12:58

Runtime 439 ms | Beats 33.40%

Memory 185.73 MB | Beats 25.58%

Code | C++

```
class Solution {
public:
```

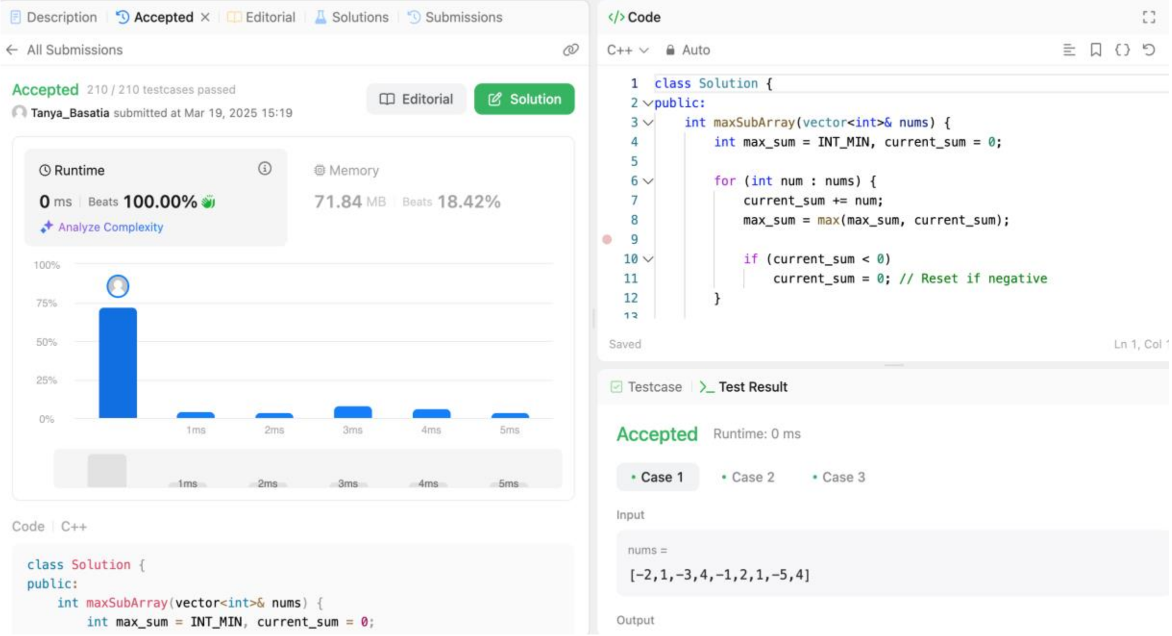
```
class Solution {
public:
    void merge(vector<int>& nums, int left, int mid, int right) {
        int n1 = mid - left + 1;
        int n2 = right - mid;

        vector<int> leftArr(n1), rightArr(n2);

        for (int i = 0; i < n1; i++)
            leftArr[i] = nums[left + i];
        for (int j = 0; j < n2; j++)
            rightArr[j] = nums[mid + 1 + j];

        int i = 0, j = 0, k = left;
        while (i < n1 && j < n2) {
            if (leftArr[i] <= rightArr[j]) {
                nums[k++] = leftArr[i++];
            } else {
```

Accepted Runtime: 0 ms

4. Maximum Subarray

```
class Solution {
public:
```

```
int maxSubArray(vector<int>& nums) {
int max_sum = INT_MIN, current_sum = 0;
for (int num : nums) {
current_sum += num;
max_sum = max(max_sum, current_sum);
if (current_sum < 0)
current_sum = 0; // Reset if negative
}
return max_sum;
}
};
```



Accepted 210 / 210 testcases passed

Tanya_Basatia submitted at Mar 19, 2025 15:19

0 ms | Beats 100.00%

71.84 MB | Beats 18.42%

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int max_sum = INT_MIN, current_sum = 0;
```

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int max_sum = INT_MIN, current_sum = 0;

        for (int num : nums) {
            current_sum += num;
            max_sum = max(max_sum, current_sum);

            if (current_sum < 0)
                current_sum = 0; // Reset if negative
        }
```

Accepted Runtime: 0 ms

Input

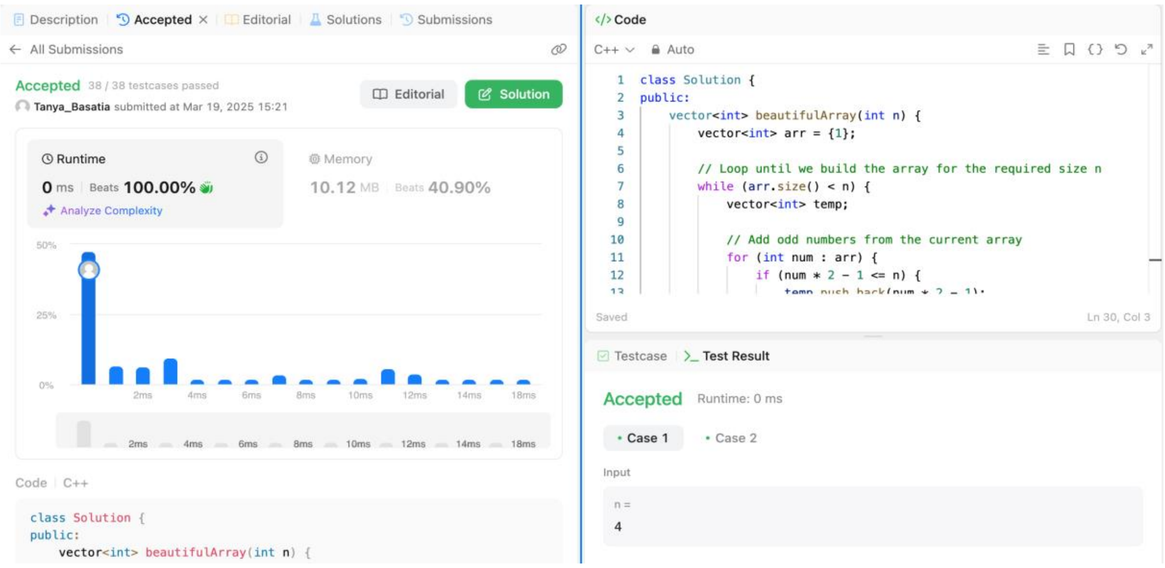
```
nums =
[-2,1,-3,4,-1,2,1,-5,4]
```

Output

5. Beautiful Array

```
class Solution {
public:
vector<int> beautifulArray(int n) {
vector<int> arr = {1};
// Loop until we build the array for the required size n
while (arr.size() < n) {
vector<int> temp;
// Add odd numbers from the current array
for (int num : arr) {
if (num * 2 - 1 <= n) {
temp.push_back(num * 2 - 1);
}
}
// Add even numbers from the current array
for (int num : arr) {
```

```
if (num * 2 <= n) {
temp.push_back(num * 2);
}
}
// Update arr to be the new temp array
arr = temp;
}
return arr;
}
};
```



Accepted 38 / 38 testcases passed

Tanya_Basatia submitted at Mar 19, 2025 15:21

Runtime

0 ms | Beats 100.00%

Memory

10.12 MB | Beats 40.90%

Code | C++

```
class Solution {
public:
    vector<int> beautifulArray(int n) {
```

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

        // Loop until we build the array for the required size n
        while (arr.size() < n) {
            vector<int> temp;

            // Add odd numbers from the current array
            for (int num : arr) {
                if (num * 2 - 1 <= n) {
                    temp.push_back(num * 2 - 1);
```

Saved

Ln 30, Col 3

Accepted Runtime: 0 ms

Input

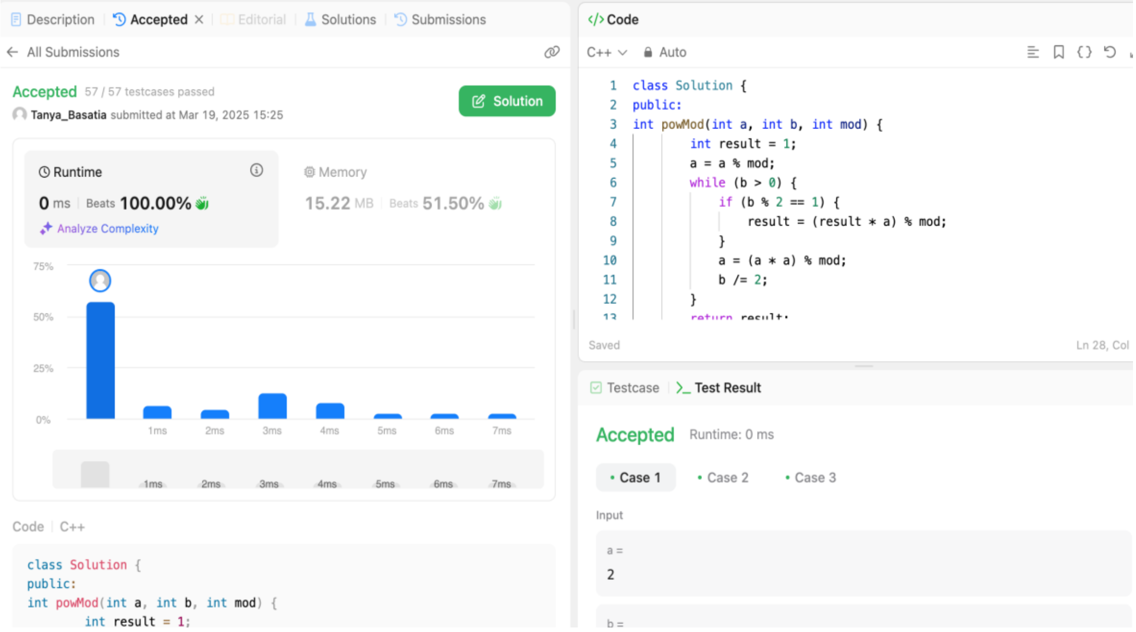
n =

4

6. Super Pow

```
class Solution {
public:
int powMod(int a, int b, int mod) {
int result = 1;
a = a % mod;
while (b > 0) {
if (b % 2 == 1) {
result = (result * a) % mod;
}
a = (a * a) % mod;
b /= 2;
}
return result;
}

int superPow(int a, vector<int>& b) {
int mod = 1337;
a = a % mod;
int result = 1;
for (int i = 0; i < b.size(); i++) {
```

```
result = (powMod(result, 10, mod) * powMod(a, b[i], mod)) % mod;
}
return result;
}
};
```



Accepted 57 / 57 testcases passed

Tanya_Basatia submitted at Mar 19, 2025 15:25

Runtime 0 ms | Beats 100.00%

Memory 15.22 MB | Beats 51.50%

Code | C++

```
class Solution {
public:
int powMod(int a, int b, int mod) {
        int result = 1;
```

```
class Solution {
public:
int powMod(int a, int b, int mod) {
        int result = 1;
        a = a % mod;
        while (b > 0) {
            if (b % 2 == 1) {
                result = (result * a) % mod;
            }
            a = (a * a) % mod;
            b /= 2;
        }
        return result;
```

Saved

Ln 28, Col

Accepted Runtime: 0 ms

Input

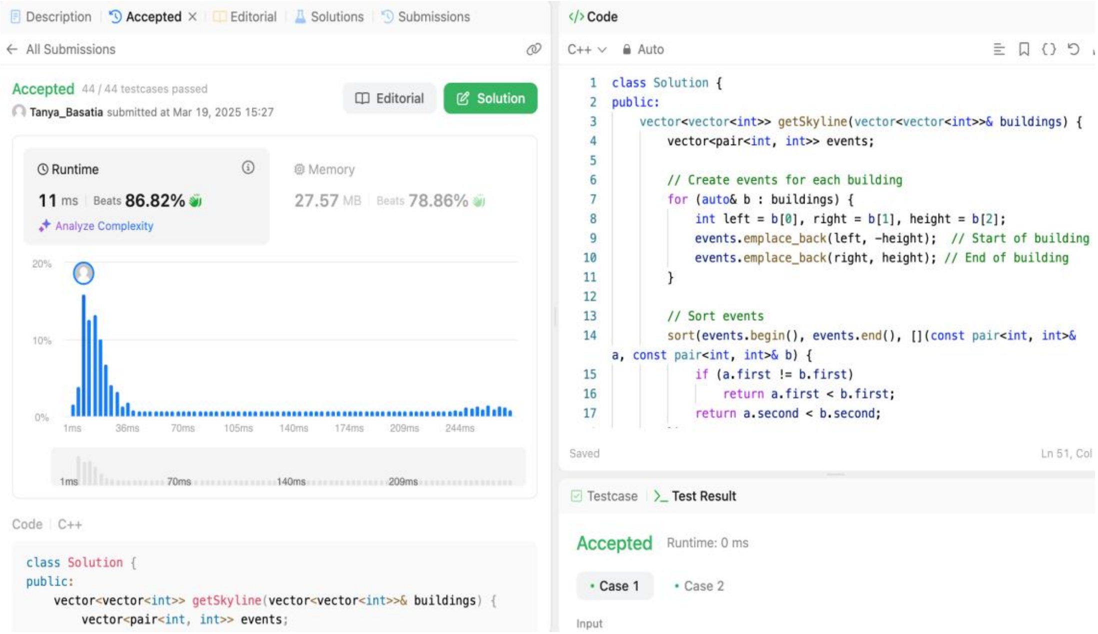
a =

2

b =

7. The Skyline Problem

```
class Solution {
public:
vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
vector<pair<int, int>> events;
// Create events for each building
for (auto& b : buildings) {
int left = b[0], right = b[1], height = b[2];
events.emplace_back(left, -height); // Start of building
events.emplace_back(right, height); // End of building
}
// Sort events
sort(events.begin(), events.end(), [](const pair<int, int>& a, const pair<int, int>& b) {
if (a.first != b.first)
return a.first < b.first;
return a.second < b.second;
});
// Max heap to store active building heights
multiset<int> heights = {0};
int prevMaxHeight = 0;
vector<vector<int>> result;
// Process all events
for (auto& event : events) {
int x = event.first, h = event.second;
if (h < 0) {
// Start of a building: add height
heights.insert(-h);
} else {
// End of a building: remove height
heights.erase(heights.find(h));
}
int currMaxHeight = *heights.rbegin();
if (currMaxHeight != prevMaxHeight) {
result.push_back({x, currMaxHeight});
prevMaxHeight = currMaxHeight;
}
}
return result;
}
};
```



Accepted 44 / 44 testcases passed

Tanya_Basatia submitted at Mar 19, 2025 15:27

Runtime

11 ms | Beats 86.82%

Memory

27.57 MB | Beats 78.86%

Code | C++

```
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<pair<int, int>> events;
```

```
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<pair<int, int>> events;

        // Create events for each building
        for (auto& b : buildings) {
            int left = b[0], right = b[1], height = b[2];
            events.emplace_back(left, -height);  // Start of building
            events.emplace_back(right, height); // End of building
        }

        // Sort events
        sort(events.begin(), events.end(), [](const pair<int, int>&
a, const pair<int, int>& b) {
            if (a.first != b.first)
                return a.first < b.first;
            return a.second < b.second;
```

Accepted Runtime: 0 ms