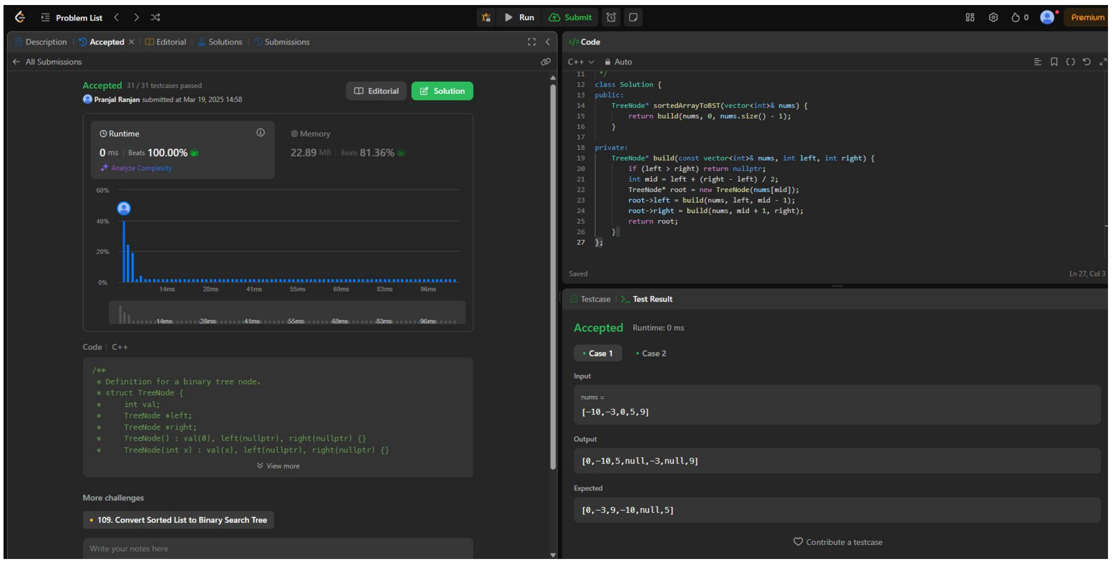
PRANJAL RANJAN ,22BCS13706
ASSIGNMENT 6

108. CONVERT SORTED ARRAY TO BINARY SEARCH TREE

```
/**
 * 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(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
 * };
 */
class Solution {
public:
    TreeNode* sortedArrayToBST(vector<int>& nums) {
        return build(nums, 0, nums.size() - 1);
    }

private:
    TreeNode* build(const vector<int>& nums, int left, int right) {
        if (left > right) return nullptr;
        int mid = left + (right - left) / 2;
        TreeNode* root = new TreeNode(nums[mid]);
        root->left = build(nums, left, mid - 1);
        root->right = build(nums, mid + 1, right);
        return root;
    }
};
```



Accepted 31 / 31 testcases passed
Pranjal Ranjan submitted at Mar 19, 2025 14:58

Runtime
0 ms | Beats 100.00%
Analyze Complexity

Memory
22.89 MB | Beats 81.36%

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) {}
```

View more

More challenges
109. Convert Sorted List to Binary Search Tree

Write your notes here

```
 */
class Solution {
public:
    TreeNode* sortedArrayToBST(vector<int>& nums) {
        return build(nums, 0, nums.size() - 1);
    }

private:
    TreeNode* build(const vector<int>& nums, int left, int right) {
        if (left > right) return nullptr;
        int mid = left + (right - left) / 2;
        TreeNode* root = new TreeNode(nums[mid]);
        root->left = build(nums, left, mid - 1);
        root->right = build(nums, mid + 1, right);
        return root;
    }
};
```

Saved   Ln 27, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1   Case 2

Input
nums =
[-10,-3,0,5,9]

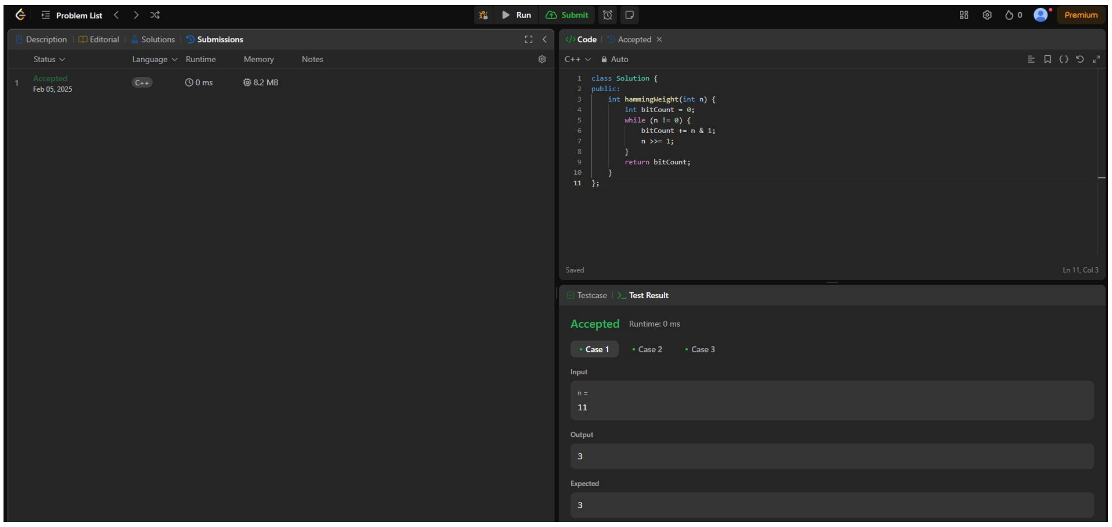
Output
[0,-10,5,null,-3,null,9]

Expected
[0,-3,9,-10,null,5]

Contribute a testcase

191. NUMBER OF 1 BITS

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



```
Status    Language  Runtime  Memory   Notes
1 Accepted     C++       0 ms     8.2 MB
  Feb 05, 2025
```

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

Saved   Ln 11, Col 3

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

n =
11

Output

3

Expected

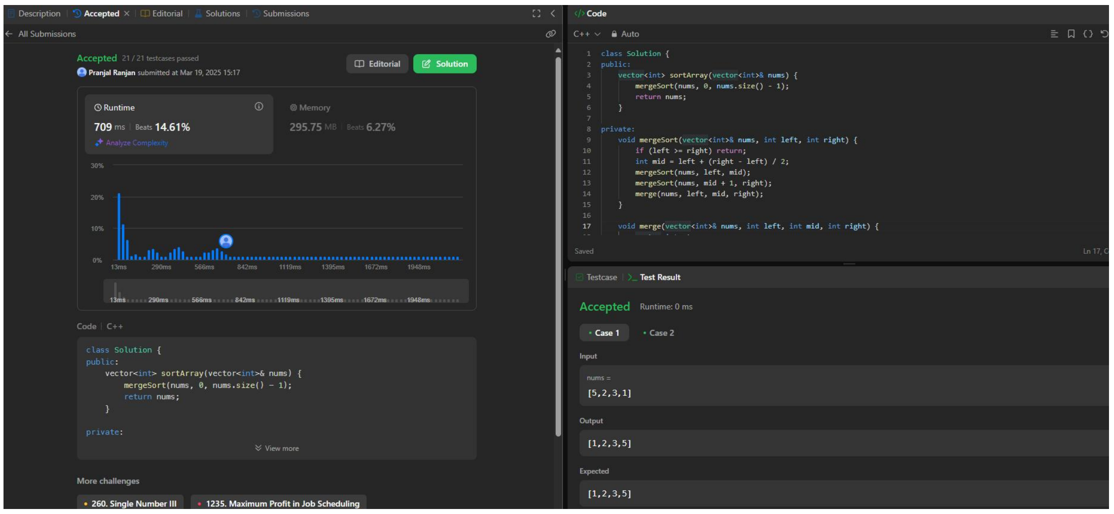
3

912. SORT AN ARRAY

```cpp
class Solution {
public:
    vector<int> sortArray(vector<int>& nums) {
        mergeSort(nums, 0, nums.size() - 1);
        return nums;
    }

private:
    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);
    }
```

```cpp
    void merge(vector<int>& nums, int left, int mid, int right) {
        vector<int> temp;
        int i = left, j = mid + 1;
        while (i <= mid && j <= right) {
            if (nums[i] <= nums[j]) temp.push_back(nums[i++]);
            else temp.push_back(nums[j++]);
        }
        while (i <= mid) temp.push_back(nums[i++]);
        while (j <= right) temp.push_back(nums[j++]);
        for (int k = left; k <= right; ++k) nums[k] = temp[k - left];
    }
};
```



Accepted 21 / 21 testcases passed

Pranjal Ranjan submitted at Mar 19, 2025 15:17

Runtime
709 ms | Beats 14.61%

Memory
295.75 MB | Beats 6.27%

Code | C++

```
class Solution {
public:
    vector<int> sortArray(vector<int>& nums) {
        mergeSort(nums, 0, nums.size() - 1);
        return nums;
    }

private:
```

More challenges

260. Single Number III

1235. Maximum Profit in Job Scheduling

```
class Solution {
public:
    vector<int> sortArray(vector<int>& nums) {
        mergeSort(nums, 0, nums.size() - 1);
        return nums;
    }

private:
    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);
    }

    void merge(vector<int>& nums, int left, int mid, int right) {
```

Accepted Runtime: 0 ms

Input

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

Output

```
[1,2,3,5]
```

Expected

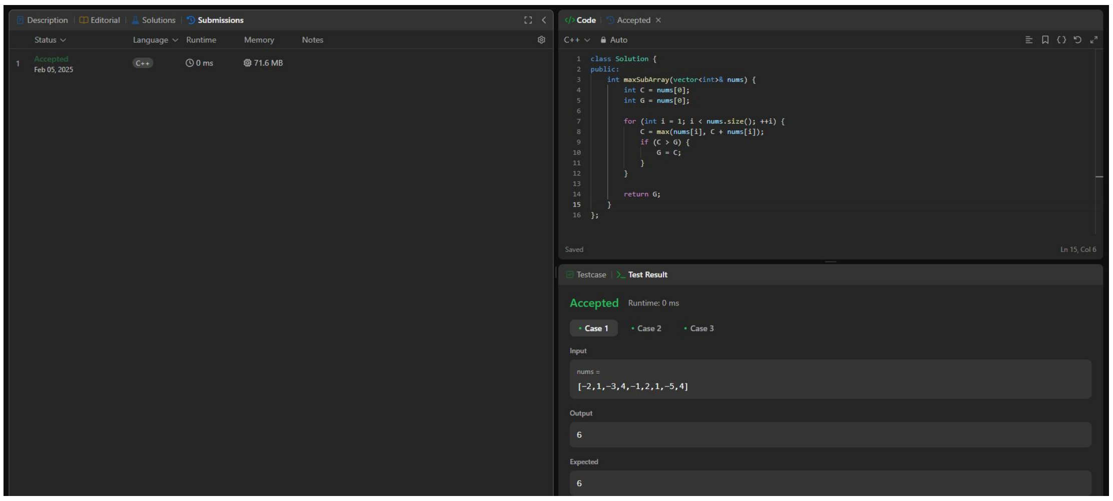
```
[1,2,3,5]
```

53.MAXIMUM SUBARRAY

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int C = nums[0];
        int G = nums[0];

        for (int i = 1; i < nums.size(); ++i) {
            C = max(nums[i], C + nums[i]);
            if (C > G) {
                G = C;
            }
        }
```

```
        return G;
    }
};
```



Status	Language	Runtime	Memory	Notes
1 Accepted Feb 05, 2025	C++	0 ms	71.6 MB	

```cpp
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int C = nums[0];
        int G = nums[0];

        for (int i = 1; i < nums.size(); ++i) {
            C = max(nums[i], C + nums[i]);
            if (C > G) {
                G = C;
            }
        }

        return G;
    }
};
```

Saved

Ln 15, Col 6

Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

Input

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

Output

```
6
```

Expected

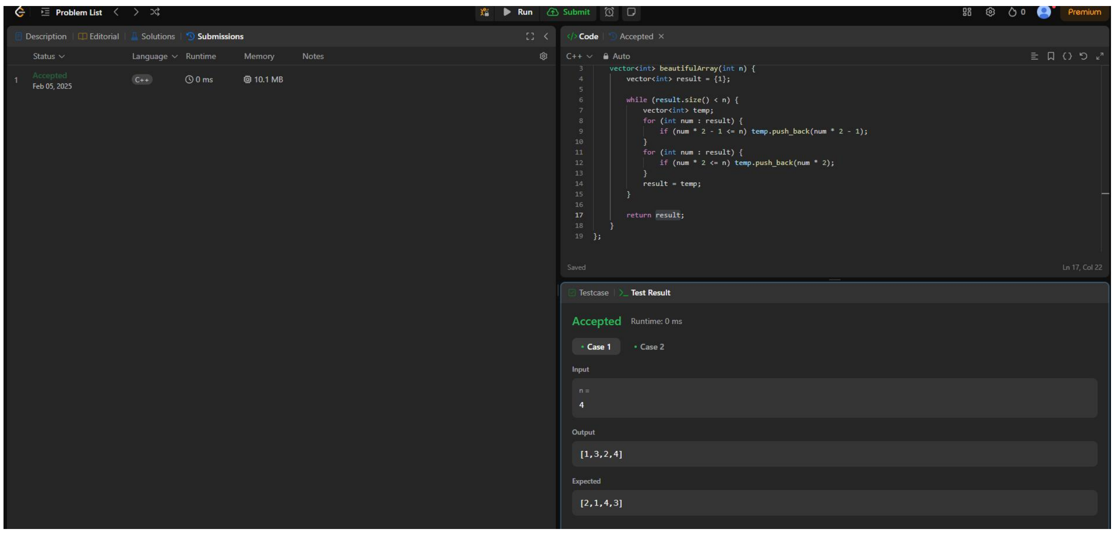
```
6
```

932. BEAUTIFUL ARRAY

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

        while (result.size() < n) {
            vector<int> temp;
            for (int num : result) {
                if (num * 2 - 1 <= n) temp.push_back(num * 2 - 1);
            }
            for (int num : result) {
                if (num * 2 <= n) temp.push_back(num * 2);
            }
            result = temp;
        }

        return result;
    }
};
```



Status	Language	Runtime	Memory	Notes
Accepted Feb 05, 2025	C++	0 ms	10.1 MB	

```cpp
    vector<int> beautifulArray(int n) {
        vector<int> result = {1};

        while (result.size() < n) {
            vector<int> temp;
            for (int num : result) {
                if (num * 2 - 1 <= n) temp.push_back(num * 2 - 1);
            }
            for (int num : result) {
                if (num * 2 <= n) temp.push_back(num * 2);
            }
            result = temp;
        }

        return result;
    }
};
```

Saved Ln 17, Col 22

Accepted Runtime: 0 ms

Case 1 Case 2

Input

```
n =
4
```

Output

```
[1,3,2,4]
```

Expected

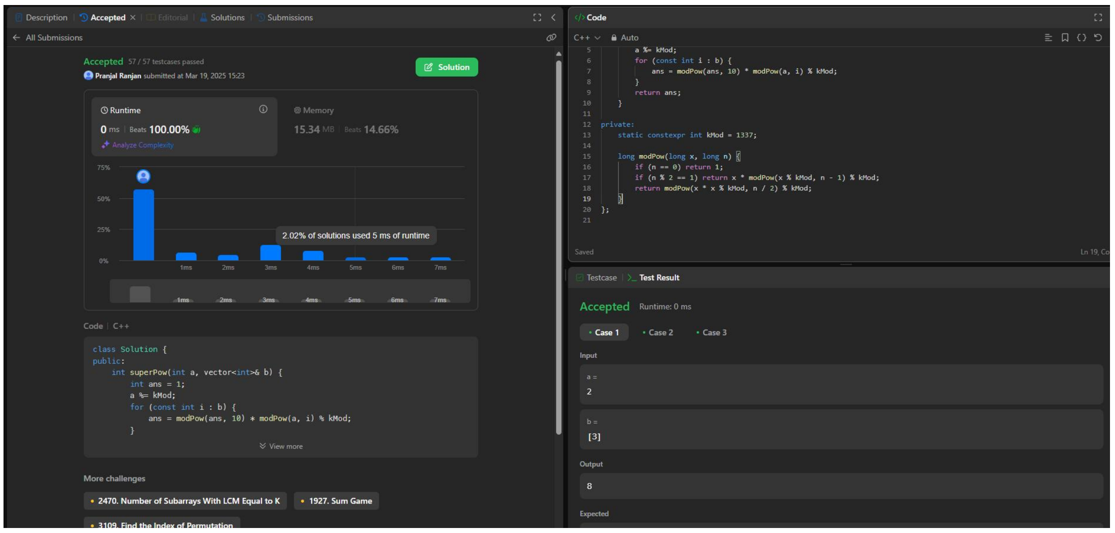
```
[2,1,4,3]
```

372. SUPER POW

```cpp
class Solution {
public:
    int superPow(int a, vector<int>& b) {
        int ans = 1;
        a %= kMod;
        for (const int i : b) {
            ans = modPow(ans, 10) * modPow(a, i) % kMod;
        }
        return ans;
    }

private:
    static constexpr int kMod = 1337;
```

```cpp
    long modPow(long x, long n) {
        if (n == 0) return 1;
        if (n % 2 == 1) return x * modPow(x % kMod, n - 1) % kMod;
        return modPow(x * x % kMod, n / 2) % kMod;
    }
};
```



Accepted 57 / 57 testcases passed

Pranjal Ranjan submitted at Mar 19, 2025 15:23

Runtime
0 ms | Beats 100.00%

Memory
15.34 MB | Beats 14.66%

2.02% of solutions used 5 ms of runtime

Code | C++

```
class Solution {
public:
    int superPow(int a, vector<int>& b) {
        int ans = 1;
        a %= kMod;
        for (const int i : b) {
            ans = modPow(ans, 10) * modPow(a, i) % kMod;
        }
```

More challenges

- 2470. Number of Subarrays With LCM Equal to K
- 1927. Sum Game
- 3109. Find the Index of Permutation

```
        a %= kMod;
        for (const int i : b) {
            ans = modPow(ans, 10) * modPow(a, i) % kMod;
        }
        return ans;
    }

private:
    static constexpr int kMod = 1337;

    long modPow(long x, long n) {
        if (n == 0) return 1;
        if (n % 2 == 1) return x * modPow(x % kMod, n - 1) % kMod;
        return modPow(x * x % kMod, n / 2) % kMod;
    }
};
```

Saved

Ln 19, Co

Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

Input

a =
2

b =
[3]

Output

8

Expected

218. THE SKYLINE PROBLEM

```
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        if (buildings.empty()) return {};
        return divideAndConquer(buildings, 0, buildings.size() - 1);
    }

private:
    vector<vector<int>> divideAndConquer(vector<vector<int>>& buildings, int
left, int right) {
        if (left == right) {
            return {{buildings[left][0], buildings[left][2]},
{buildings[left][1], 0}};
        }
        int mid = left + (right - left) / 2;
        auto leftSkyline = divideAndConquer(buildings, left, mid);
        auto rightSkyline = divideAndConquer(buildings, mid + 1, right);
        return mergeSkylines(leftSkyline, rightSkyline);
    }
```

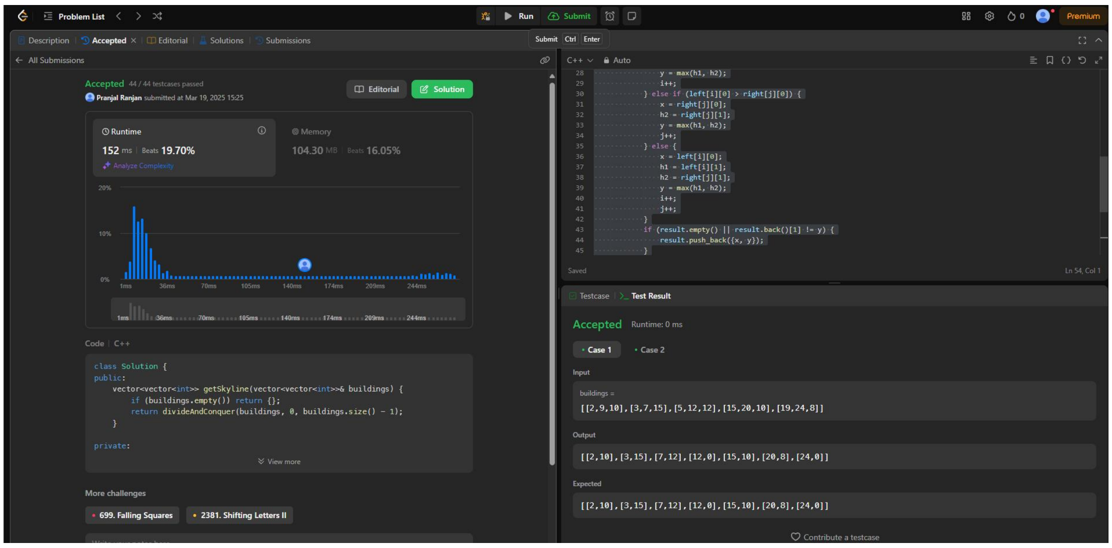
```
    vector<vector<int>> mergeSkylines(vector<vector<int>>& left,
vector<vector<int>>& right) {
        vector<vector<int>> result;
        int h1 = 0, h2 = 0, x = 0, y = 0;
        size_t i = 0, j = 0;
```

```
        while (i < left.size() && j < right.size()) {
            if (left[i][0] < right[j][0]) {
                x = left[i][0];
                h1 = left[i][1];
                y = max(h1, h2);
                i++;
            } else if (left[i][0] > right[j][0]) {
                x = right[j][0];
```

```cpp
                h2 = right[j][1];
                y = max(h1, h2);
                j++;
            } else {
                x = left[i][0];
                h1 = left[i][1];
                h2 = right[j][1];
                y = max(h1, h2);
                i++;
                j++;
            }
            if (result.empty() || result.back()[1] != y) {
                result.push_back({x, y});
            }
        }

        while (i < left.size()) result.push_back(left[i++]);
        while (j < right.size()) result.push_back(right[j++]);

        return result;
    }
};
```



Accepted 44 / 44 testcases passed
Pranjal Ranjan submitted at Mar 19, 2025 15:25

Runtime
152 ms | Beats 19.70%

Memory
104.30 MB | Beats 16.05%

Code | C++

```cpp
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        if (buildings.empty()) return {};
        return divideAndConquer(buildings, 0, buildings.size() - 1);
    }

private:
```

More challenges
- 699. Falling Squares
- 2381. Shifting Letters II

C++ Auto

```cpp
                y = max(h1, h2);
                i++;
            } else if (left[i][0] > right[j][0]) {
                x = right[j][0];
                h2 = right[j][1];
                y = max(h1, h2);
                j++;
            } else {
                x = left[i][0];
                h1 = left[i][1];
                h2 = right[j][1];
                y = max(h1, h2);
                i++;
                j++;
            }
            if (result.empty() || result.back()[1] != y) {
                result.push_back({x, y});
            }
```

Saved   Ln 54, Col 1

Accepted Runtime: 0 ms

Input
buildings =
```
[[2,9,10],[3,7,15],[5,12,12],[15,20,10],[19,24,8]]
```

Output
```
[[2,10],[3,15],[7,12],[12,0],[15,10],[20,8],[24,0]]
```

Expected
```
[[2,10],[3,15],[7,12],[12,0],[15,10],[20,8],[24,0]]
```