



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 6

Student Name: Mohd Arshad

Branch: CSE

Semester: 6TH

Subject Name: AP-II

UID: 22BCS10091

Section/Group: FL_IOT_602

Date of Performance: 19/03/25

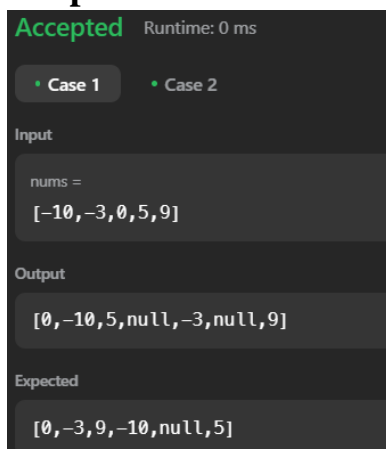
Subject Code: 22CSP-351

Q1: Convert Sorted Array to Binary Search Tree

Code:

```
#include <bits/stdc++.h>
using namespace std;
class Solution {
public:
    TreeNode* sortedArrayToBST(vector<int>& nums) {
        return buildBST(nums, 0, nums.size() - 1);
    }
private:
    TreeNode* buildBST(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 = buildBST(nums, left, mid - 1);
        root->right = buildBST(nums, mid + 1, right);
        return root;
    }
};
```

Output:





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Q2: Number of 1 Bits

Code:

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

Output:

A screenshot of a code execution interface. At the top, the word 'Accepted' is written in green, followed by 'Runtime: 0 ms'. Below this, there are three tabs labeled 'Case 1', 'Case 2', and 'Case 3', with 'Case 1' being the active tab. Under the 'Input' section, there is a text box containing 'n =' and '11'. Under the 'Output' section, there is a text box containing '3'. Under the 'Expected' section, there is a text box containing '3'.

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

n =
11

Output

3

Expected

3



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Q3: Sort an Array

Code:

```
#include <bits/stdc++.h>
using namespace std;

class Solution {
public:
    vector<int> sortArray(vector<int>& nums) {
        sort(nums.begin(), nums.end());
        return nums;
    }
};
```

Output:

A screenshot of a coding platform interface. At the top, it says 'Accepted' in green and 'Runtime: 0 ms' in white. Below this, there are two tabs: 'Case 1' and 'Case 2', both with a green dot indicating they are active. Under the 'Input' section, it shows 'nums =' followed by '[5,2,3,1]'. Under the 'Output' section, it shows '[1,2,3,5]'. Under the 'Expected' section, it also shows '[1,2,3,5]'. The background is dark grey.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Q4: Maximum Subarray

Code:

```
class Solution {  
public:  
    int maxSubArray(vector<int>& nums) {  
        int maxSum = nums[0], currentSum = nums[0];  
        for (int i = 1; i < nums.size(); ++i) {  
            currentSum = max(nums[i], currentSum + nums[i]);  
            maxSum = max(maxSum, currentSum);  
        }  
        return maxSum;  
    }  
};
```

Output:

A screenshot of a code execution interface with a dark background. At the top left, the word 'Accepted' is written in green, followed by 'Runtime: 0 ms' in white. Below this, there are three tabs labeled 'Case 1', 'Case 2', and 'Case 3', each preceded by a small green dot. Under the 'Case 1' tab, the 'Input' section shows 'nums =' followed by the array '[-2, 1, -3, 4, -1, 2, 1, -5, 4]'. The 'Output' section shows the number '6'. The 'Expected' section also shows the number '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



Q5: Beautiful Array

Code:

```
class Solution {
public:
    vector<int> beautifulArray(int n) {
        vector<int> result = {1};
        while (result.size() < n) {
            vector<int> temp;
            for (int x : result) {
                if (x * 2 - 1 <= n) temp.push_back(x * 2 - 1);
            }
            for (int x : result) {
                if (x * 2 <= n) temp.push_back(x * 2);
            }
            result = temp;
        }
        return result;
    }
};
```

Output:

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

n =
4

Output

[1,3,2,4]

Expected

[2,1,4,3]



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

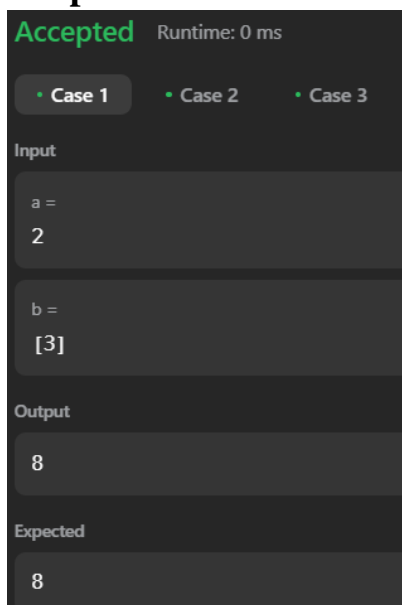
Discover. Learn. Empower.

Q6: Super Pow

Code:

```
class Solution {
public:
    int superPow(int a, vector<int>& b) {
        const int MOD = 1337;
        a %= MOD;
        int result = 1;
        for (int i = b.size() - 1; i >= 0; --i) {
            result = (result * powMod(a, b[i], MOD)) % MOD;
            a = powMod(a, 10, MOD);
        }
        return result;
    }
    int powMod(int a, int b, int MOD) {
        int res = 1;
        while (b > 0) {
            if (b % 2 == 1) res = (res * a) % MOD;
            a = (a * a) % MOD;
            b /= 2;
        }
        return res;
    }
};
```

Output:




Q7: The Skyline Problem

Code:

```
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<vector<int>> result;
        vector<pair<int, int>> heights;
        for (auto& building : buildings) {
            heights.push_back({building[0], -building[2]});
            heights.push_back({building[1], building[2]}); }
        sort(heights.begin(), heights.end());
        multiset<int> activeHeights{0};
        int prevMaxHeight = 0;
        for (auto& height : heights) {
            if (height.second < 0) {
                activeHeights.insert(-height.second);
            } else {
                activeHeights.erase(activeHeights.find(height.second)); }
            int currentMaxHeight = *activeHeights.rbegin();

            if (currentMaxHeight != prevMaxHeight) {
                result.push_back({height.first, currentMaxHeight});
                prevMaxHeight = currentMaxHeight;
            }
        }
        return result; }
};
```

Output:



The screenshot shows a code execution interface with a dark background. At the top, it says "Accepted" in green and "Runtime: 0 ms". Below this, there are two tabs: "Case 1" (selected) and "Case 2". Under the "Case 1" tab, there is an "Input" section with the text "buildings =" followed by a vector of vectors: "[[2,9,10], [3,7,15], [5,12,12], [15,20,10], [19,24,8]]". Below the input is an "Output" section showing the result: "[[2,10], [3,15], [7,12], [12,0], [15,10], [20,8], [24,0]]". At the bottom, there is an "Expected" section showing the same result: "[[2,10], [3,15], [7,12], [12,0], [15,10], [20,8], [24,0]]".