



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Assingment -6

**Student Name:** Subhojit Ghosh

**UID:** 22BCS15368

**Branch:** CSE

**Section/Group:** FL-602-A

**Semester:** 6

**Date of Performance:** 18 march 25

**Subject Name:** Advanced Programming

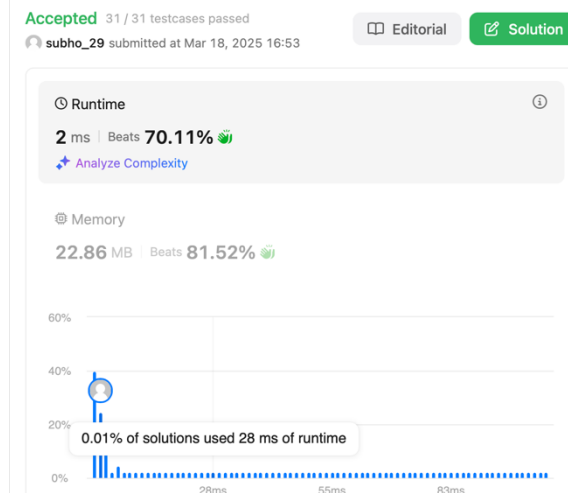
**Subject Code:** 22CSH-359

### 1. Convert Sorted Array to Binary Search Tree

```
#include <vector>
using namespace std;

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

private:
    TreeNode* helper(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 = helper(nums, left, mid - 1);
        root->right = helper(nums, mid + 1, right);
        return root;
    }
};
```





# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## 2. Number of Bits

```
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;
    }
};
```

Accepted 598 / 598 testcases passed

subho\_29 submitted at Mar 18, 2025 16:57

Editorial

Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

8.30 MB | Beats 47.49%



## 3. Sort an Array

```
class Solution {
public:
    int partition(vector<int>& arr, int l, int r) {
        if (l >= r) return -1;
        int n=r-1;
        int pivot=l+rand()%n;
        swap(arr[l],arr[pivot]);
        int i=l+1;
        for(int j=l+1;j<r;j++){
            if(arr[j]<arr[l]){
                swap(arr[i],arr[j]);
                i++;
            }
        }
        swap(arr[l],arr[i-1]);
        return i-1;
    }
    void QuickSort(vector<int>& nums, int l, int r) {
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        if(l>=r){
            return;
        }
        int pivot=partition(nums,l,r);
        QuickSort(nums,l,pivot);
        QuickSort(nums,pivot+1,r);
    }
    vector<int> sortArray(vector<int>& nums) {
        int n=nums.size();
        QuickSort(nums,0,n);
        return nums;
    }
};
```

Accepted 21 / 21 testcases passed

subho\_29 submitted at Mar 18, 2025 17:02

Editorial

Solution

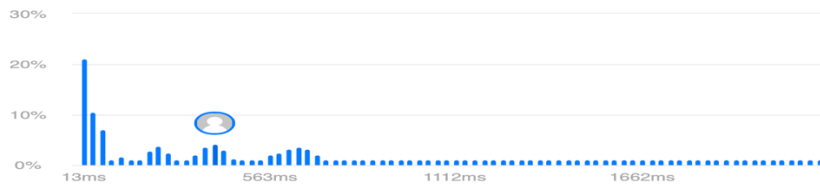
Runtime

399 ms | Beats 38.80%

Analyze Complexity

Memory

185.60 MB | Beats 31.29%



## 4. Maximum Subarray

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

        for (int n : nums) {
            if (total < 0) {
                total = 0;
            }

            total += n;
            res = max(res, total);
        }

        return res;
    }
};
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Accepted 210 / 210 testcases passed

subho\_29 submitted at Mar 18, 2025 17:06

Editorial

Solution

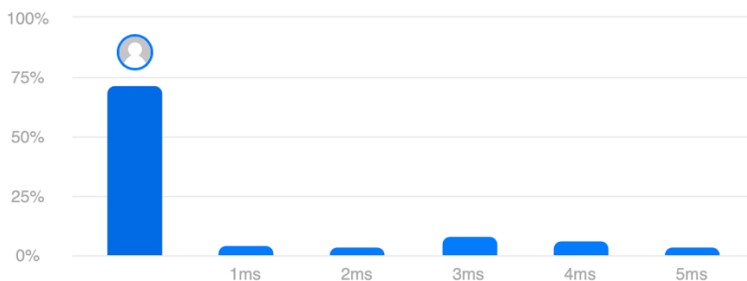
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

71.69 MB | Beats 81.06%



## 5. Beautiful Array

```
class Solution {
public:
    int partition(vector<int> &v, int start, int end, int mask)
    {
        int j = start;
        for(int i = start; i <= end; i++)
        {
            if((v[i] & mask) != 0)
            {
                swap(v[i], v[j]);
                j++;
            }
        }
        return j;
    }

    void sort(vector<int> & v, int start, int end, int mask)
    {
        if(start >= end) return;
        int mid = partition(v, start, end, mask);
        sort(v, start, mid - 1, mask << 1);
        sort(v, mid, end, mask << 1);
    }
}
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
vector<int> beautifulArray(int N) {  
    vector<int> ans;  
    for(int i = 0; i < N; i++) ans.push_back(i + 1);  
    sort(ans, 0, N - 1, 1);  
    return ans;  
}  
};
```

Accepted 38 / 38 testcases passed

subho\_29 submitted at Mar 18, 2025 17:10

Editorial

Solution

Runtime

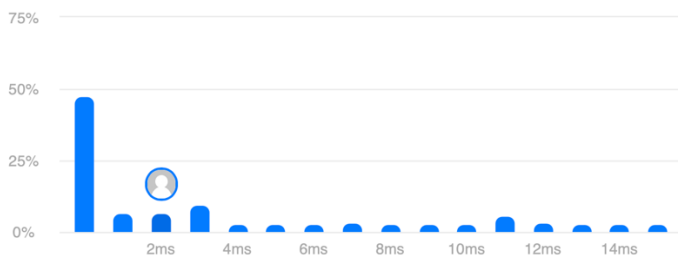
2 ms | Beats 45.73%

Analyze Complexity

Memory

9.35 MB | Beats 97.29%

Analyze Complexity



## 6. Super pow

```
class Solution {  
    const int base = 1337;  
    int powmod(int a, int k) //a^k mod 1337 where 0 <= k <= 10  
    {  
        a %= base;  
        int result = 1;  
        for (int i = 0; i < k; ++i)  
            result = (result * a) % base;  
        return result;  
    }  
public:  
    int superPow(int a, vector<int>& b) {  
        if (b.empty()) return 1;  
        int last_digit = b.back();  
        b.pop_back();  
        return powmod(superPow(a, b), 10) * powmod(a, last_digit) % base;  
    }  
};
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

};

Accepted 57 / 57 testcases passed

subho\_29 submitted at Mar 18, 2025 17:12

[Solution](#)

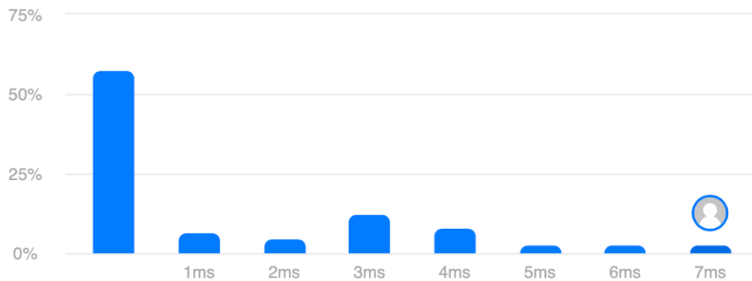
Runtime

7 ms | Beats 7.46%

[Analyze Complexity](#)

Memory

15.30 MB | Beats 51.99%



## 7. [The Skyline Problem](#)

```
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<vector<int>> ans;
        multiset<int> pq{0};

        vector<pair<int, int>> points;

        for(auto b: buildings){
            points.push_back({b[0], -b[2]});
            points.push_back({b[1], b[2]});
        }

        sort(points.begin(), points.end());

        int ongoingHeight = 0;

        // points.first = x coordinate, points.second = height
        for(int i = 0; i < points.size(); i++){
            int currentPoint = points[i].first;
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
int heightAtCurrentPoint = points[i].second;

if(heightAtCurrentPoint < 0){
    pq.insert(-heightAtCurrentPoint);
} else {
    pq.erase(pq.find(heightAtCurrentPoint));
}

// after inserting/removing heightAtI, if there's a change
auto pqTop = *pq.rbegin();
if(ongoingHeight != pqTop){
    ongoingHeight = pqTop;
    ans.push_back({currentPoint, ongoingHeight});
}

}

return ans;
};
```

Accepted 44 / 44 testcases passed

subho\_29 submitted at Mar 18, 2025 17:14

Editorial

Solution

Runtime

18 ms | Beats 55.11%

Analyze Complexity

Memory

28.95 MB | Beats 42.21%

Analyze Complexity





# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

---

Discover. Learn. Empower.



```

        leftmost_value = node->val;

        if (node->right) {
            q.push(node->right);
        }
        if (node->left) {
            q.push(node->left);
        }
    }

    return leftmost_value;
}

};

```

Accepted

subho\_29 submitted at Feb 14, 2025 13:51

Editorial

Solution

Runtime

17 ms | Beats 5.78%

Analyze Complexity

Memory

24.91 MB | Beats 63.66%



## 8. Binary Tree Maximum Path Sum

```

class Solution {
public:
    int maxPathSum(TreeNode* root) {
        int ans = INT_MIN;
        maxPathSumDownFrom(root, ans);
        return ans;
    }

private:
    int maxPathSumDownFrom(TreeNode* root, int& ans) {

```

```

    if (root == nullptr)
        return 0;
    const int l = max(0, maxPathSumDownFrom(root->left, ans));
    const int r = max(0, maxPathSumDownFrom(root->right, ans));
    ans = max(ans, root->val + l + r);
    return root->val + max(l, r);
}
};

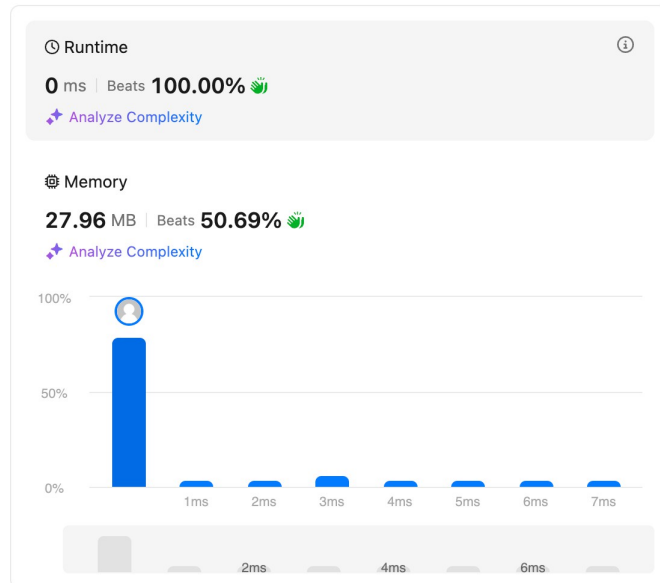
```

Accepted 96 / 96 testcases passed

subho\_29 submitted at Feb 14, 2025 15:45

Editorial

Solution



## 9. Vertical Order Traversal of a Binary Tree

```

class Solution {
public:
    vector<vector<int>> verticalTraversal(TreeNode* root) {
        map<int, map<int, multiset<int>>> nodes;
        queue<pair<TreeNode*, pair<int, int>>> q;
        q.push({root, {0, 0}});
        while (!q.empty()) {
            auto t = q.front();
            q.pop();
            TreeNode* a = t.first;
            int x = t.second.first, y = t.second.second;
            nodes[x][y].insert(a->val);
            if (a->left) {

```

```

        q.push({a->left, {x-1, y+1}});
    }
    if(a->right) {
        q.push({a->right, {x+1, y+1}});
    }

}

vector<vector<int>>ans;
for(auto p: nodes) {
    vector<int>col;
    for(auto b:p.second) {
        col.insert(col.end(), b.second.begin(), b.second.end());
    }
    ans.push_back(col);
}
return ans;
}

};

```

Accepted 34 / 34 testcases passed

subho\_29 submitted at Feb 14, 2025 15:47

Editorial

Solution

## Runtime

1 ms | Beats 60.15%

Analyze Complexity

## Memory

16.31 MB | Beats 46.72%

