



Assignment 4.

Student Name: Mannat Gupta

Branch: BE-CSE

Semester: 6th

Subject Name: Advanced Programming Lab-2

UID: 22BCS15281

Section/Group: 608-B

Date of Performance: 17/03/25

Subject Code: 22CSP-351

Aim: Divide and Conquer

1. **The Skyline Problem:** <https://leetcode.com/problems/the-skyline-problem/description/>

```
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<pair<int, int>> points;
        for (auto& b : buildings) {
            points.emplace_back(b[0], -b[2]);
            points.emplace_back(b[1], b[2]);
        }
        sort(points.begin(), points.end());

        multiset<int> heights = {0};
        vector<vector<int>> result;
        int prevHeight = 0;

        for (auto& p : points) {
            if (p.second < 0)
                heights.insert(-p.second);
            else
                heights.erase(heights.find(p.second));

            int currHeight = *heights.rbegin();
            if (currHeight != prevHeight) {
                result.push_back({p.first, currHeight});
                prevHeight = currHeight;
            }
        }
        return result;
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

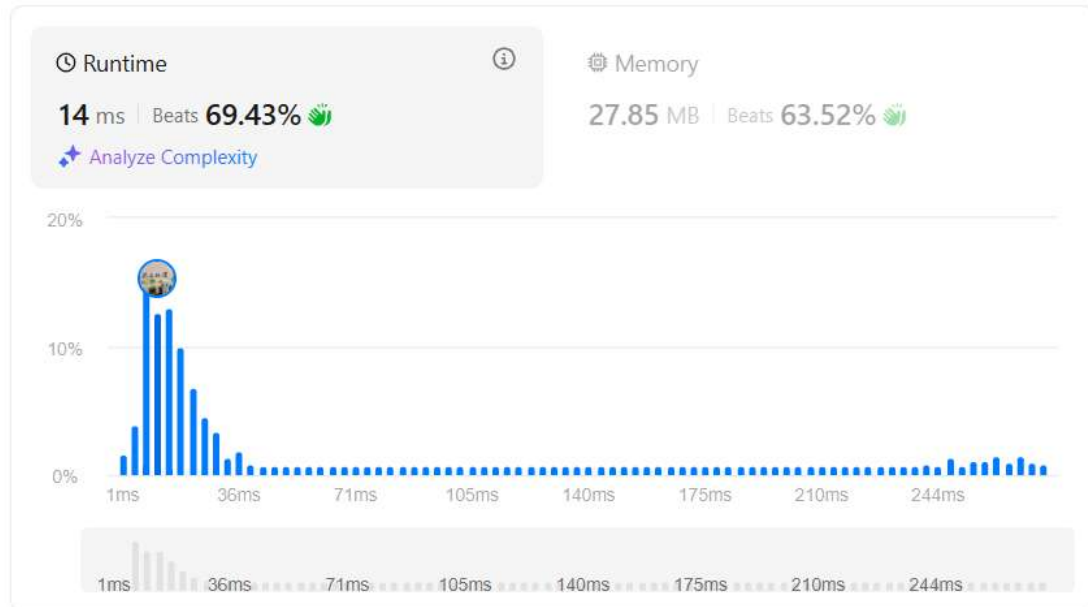
Discover. Learn. Empower.

Accepted 44 / 44 testcases passed

Mannat Gupta submitted at Mar 17, 2025 19:17

Editorial

Solution



2. Super Pow: <https://leetcode.com/problems/super-pow/description/>

```
class Solution {
public:
    const int MOD = 1337;

    int powerMod(int a, int b) {
        int res = 1;
        a %= MOD;
        while (b > 0) {
            if (b % 2 == 1)
                res = (res * a) % MOD;
            a = (a * a) % MOD;
            b /= 2;
        }
        return res;
    }

    int superPow(int a, vector<int>& b) {
        if (b.empty()) return 1;
        int lastDigit = b.back();
        b.pop_back();
        int part1 = powerMod(a, lastDigit);
        int part2 = powerMod(superPow(a, b), 10);
        return (part1 * part2) % MOD;
    }
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

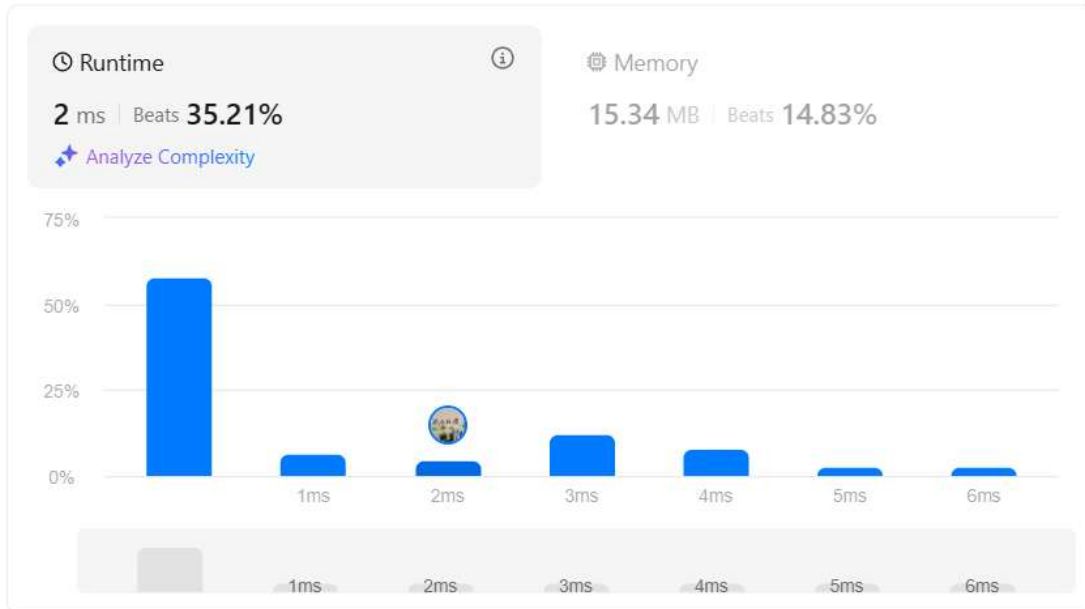
Discover. Learn. Empower.

```
};
```

Accepted 57 / 57 testcases passed

Mannat Gupta submitted at Mar 17, 2025 19:10

[Solution](#)



3. Reverse Pairs: <https://leetcode.com/problems/reverse-pairs/description/>

```
class Solution {
public:
    int mergeAndCount(vector<int>& nums, int left, int mid, int right) {
        int count = 0, j = mid + 1;
        for (int i = left; i <= mid; i++) {
            while (j <= right && (long)nums[i] > 2LL * nums[j])
                j++;
            count += (j - mid - 1);
        }

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

        return count;
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

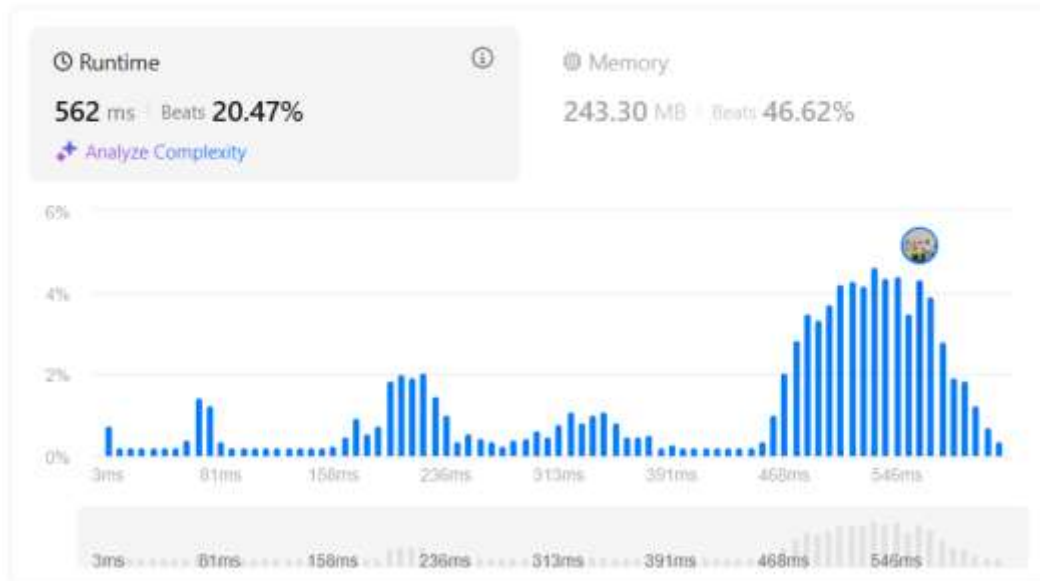
```
int mergeSort(vector<int>& nums, int left, int right) {  
    if (left >= right) return 0;  
    int mid = left + (right - left) / 2;  
    int count = mergeSort(nums, left, mid) + mergeSort(nums, mid + 1, right);  
    count += mergeAndCount(nums, left, mid, right);  
    return count;  
}  
  
int reversePairs(vector<int>& nums) {  
    return mergeSort(nums, 0, nums.size() - 1);  
}  
};
```

Accepted 140 / 140 testcases passed

Mannat Gupta submitted at Mar 17, 2025 19:19

Editorial

Solution



4. Longest Nice Substring: <https://leetcode.com/problems/longest-nice-substring/description/>

```
class Solution {  
public:  
    string longestNiceSubstring(string s) {  
        int n = s.size();  
        if (n < 2) return "";  
  
        unordered_set<char> st(s.begin(), s.end());  
  
        for (int i = 0; i < n; ++i) {  
            if (st.count(tolower(s[i])) && st.count(toupper(s[i]))) continue;  
            string left = longestNiceSubstring(s.substr(0, i));  
            string right = longestNiceSubstring(s.substr(i + 1));  
        }  
    }  
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

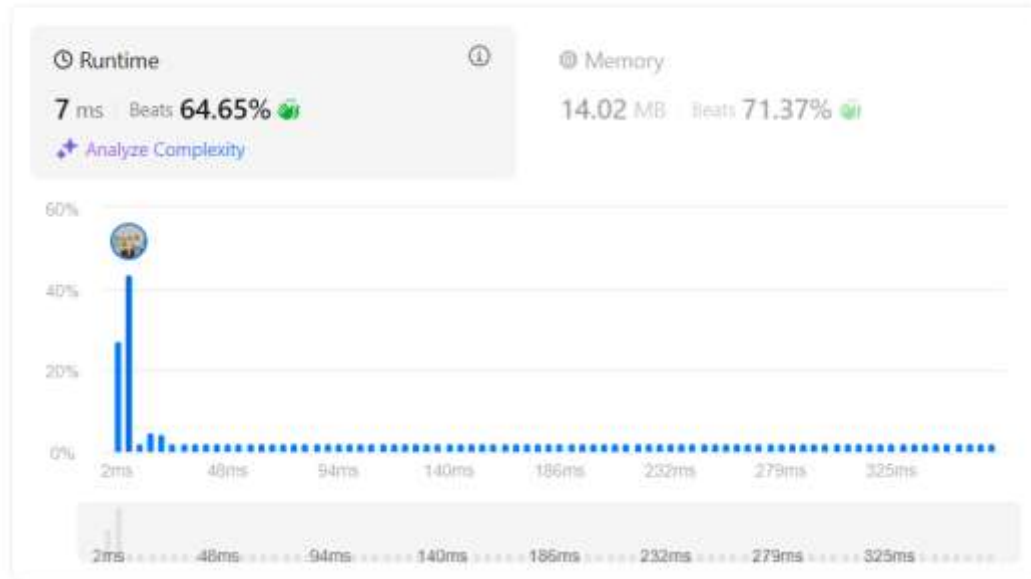
```
        return left.size() >= right.size() ? left : right;
    }

    return s;
}
};
```

Accepted 73 / 73 testcases passed

Mannat Gupta submitted at Mar 17, 2025 19:00

Solution



5. Search a 2D Matrix II: <https://leetcode.com/problems/search-a-2d-matrix-ii/description/>

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int rows = matrix.size(), cols = matrix[0].size();
        int row = 0, col = cols - 1;

        while (row < rows && col >= 0) {
            if (matrix[row][col] == target)
                return true;
            else if (matrix[row][col] > target)
                col--;
            else
                row++;
        }
        return false;
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

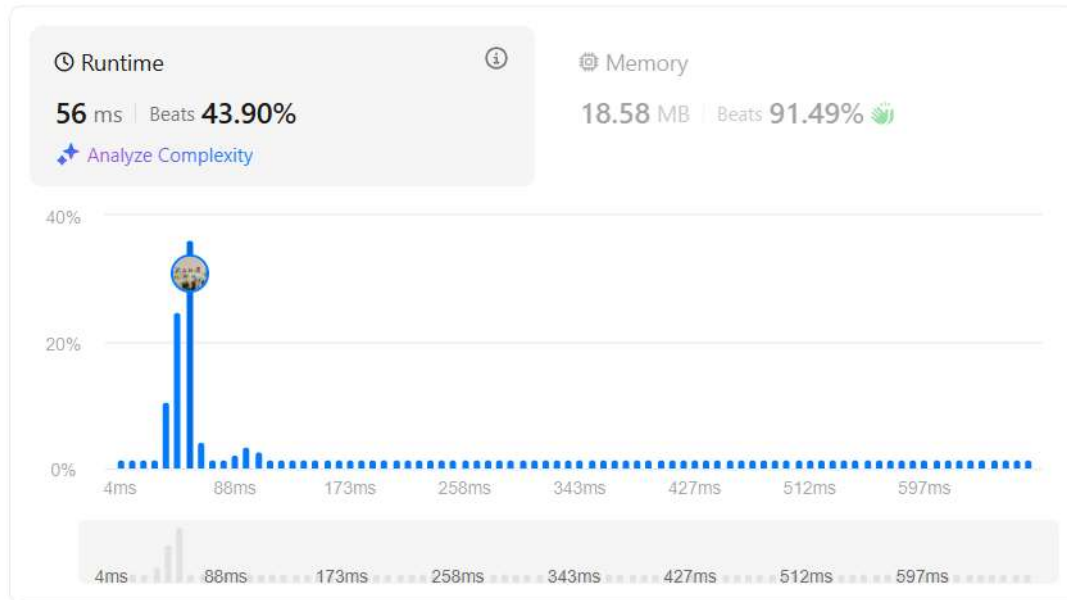
Discover. Learn. Empower.

Accepted 130 / 130 testcases passed

Mannat Gupta submitted at Mar 17, 2025 19:08

Editorial

Solution



6. Longest Increasing Subsequence II: <https://leetcode.com/problems/longest-increasing-subsequence-ii/description/>

```
class Solution {
public:
    vector<int> segTree;
    int maxVal = 1e5; // Given constraint in problem

    int query(int l, int r, int node, int start, int end) {
        if (r < start || end < l) return 0;
        if (l <= start && end <= r) return segTree[node];
        int mid = (start + end) / 2;
        return max(query(l, r, 2 * node, start, mid), query(l, r, 2 * node + 1, mid + 1,
end));
    }

    void update(int idx, int val, int node, int start, int end) {
        if (start == end) {
            segTree[node] = val;
            return;
        }
        int mid = (start + end) / 2;
        if (idx <= mid) update(idx, val, 2 * node, start, mid);
        else update(idx, val, 2 * node + 1, mid + 1, end);
        segTree[node] = max(segTree[2 * node], segTree[2 * node + 1]);
    }
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
int lengthOfLIS(vector<int>& nums, int k) {  
    segTree.assign(4 * (maxVal + 1), 0);  
    int res = 0;  
    for (int num : nums) {  
        int maxLIS = query(max(1, num - k), num - 1, 1, 1, maxVal);  
        int newLIS = maxLIS + 1;  
        update(num, newLIS, 1, 1, maxVal);  
        res = max(res, newLIS);  
    }  
    return res;  
}  
};
```

Accepted 84 / 84 testcases passed

Mannat Gupta submitted at Mar 17, 2025 19:22

Solution



7. Number of 1 Bits: <https://leetcode.com/problems/number-of-1-bits/description/>

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



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Accepted 988 / 998 testcases passed

Mannat Gupta submitted at Mar 17, 2025 19:05

Editorial

Solution



8. Reverse Bits: <https://leetcode.com/problems/reverse-bits/description/>

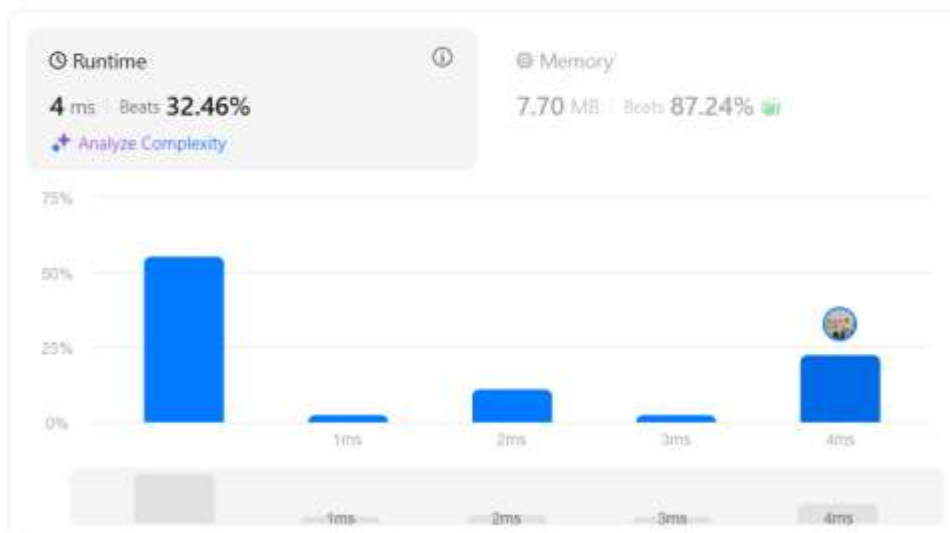
```
class Solution {  
public:  
    uint32_t reverseBits(uint32_t n) {  
        uint32_t result = 0;  
        for (int i = 0; i < 32; i++) {  
            result = (result << 1) | (n & 1);  
            n >>= 1;  
        }  
        return result;  
    }  
};
```

Accepted 600 / 600 testcases passed

Mannat Gupta submitted at Mar 17, 2025 19:02

Editorial

Solution





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

9. Beautiful Array: <https://leetcode.com/problems/beautiful-array/description/>

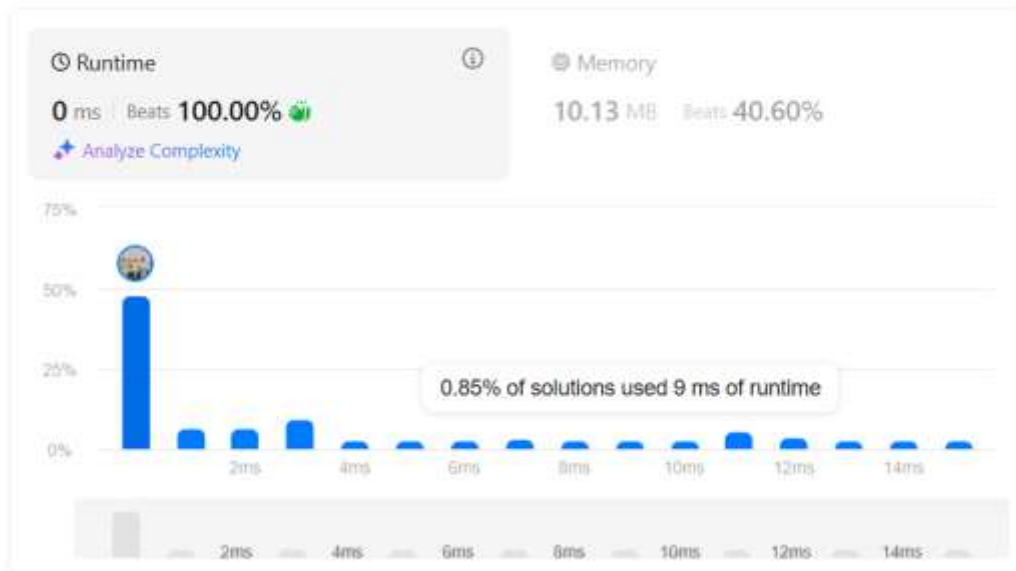
```
class Solution {
public:
    vector<int> beautifulArray(int n) {
        vector<int> res = {1};
        while (res.size() < n) {
            vector<int> temp;
            for (int num : res)
                if (2 * num - 1 <= n)
                    temp.push_back(2 * num - 1);
            for (int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

Accepted 38 / 38 testcases passed

Mannat Gupta submitted at Mar 17, 2025 19:12

Editorial

Solution



10. Maximum Subarray: <https://leetcode.com/problems/maximum-subarray/description/>

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



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
for (int i = 1; i < nums.size(); i++) {  
    currSum = max(nums[i], currSum + nums[i]);  
    maxSum = max(maxSum, currSum);  
}  
  
return maxSum;  
}  
};
```

Accepted 210 / 210 testcases passed

Mannat Gupta submitted at Mar 17, 2025 19:07

Editorial

Solution

Runtime

2 ms Beats 23.59%

Analyze Complexity

Memory

71.82 MB Beats 18.35%

Analyze Complexity

