Problem List

Run Submit Premium

Description | Accepted × | Editorial | Solutions | Submissions

All Submissions

Accepted 57 / 57 testcases passed

Anirudh Singh submitted at Mar 17, 2025 10:31

Solution

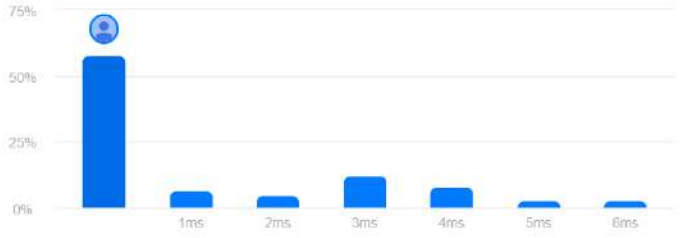
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

15.04 MB | Beats 99.10%



Code

C++ | Auto

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

    int modPower(int a, int b) {
        a %= MOD;
        int result = 1;
        for (int i = 0; i < b; ++i) {
            result = (result * a) % MOD;
        }
        return result;
    }

    int superPow(int a, vector<int>& b) {
        if (b.empty()) return 1;

        int lastDigit = b.back();
        b.pop_back();

        int part1 = modPower(a, lastDigit);
        int part2 = modPower(superPow(a, b), 10);

        return (part1 * part2) % MOD;
    }
};
```

Saved

Ln 25, Col 3

Description | Accepted × | Editorial | Solutions | Submissions

Submit Ctrl Enter

All Submissions

Accepted 73 / 73 testcases passed

Anirudh Singh submitted at Mar 17, 2025 10:31

Solution

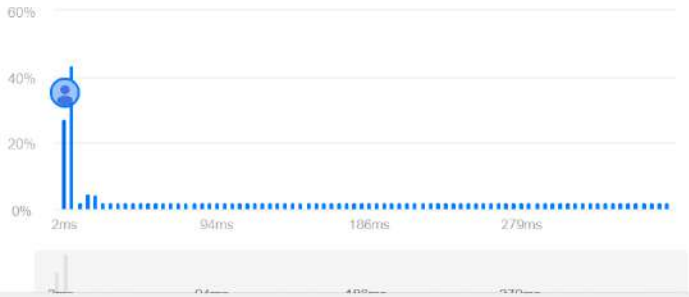
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

8.10 MB | Beats 99.70%



C++ | Auto

```cpp
class Solution {
public:
    string longestNiceSubstring(string s) {
        int n = s.length();
        int strindex = 0, maxlength = 0;
        for(int i=0;i<=n;i++){
            int lower = 0, upper = 0;
            for(int j=i;j<n;j++){
                char c = s[j];
                if (islower(c))
                    lower |= 1 << (c - 'a');
                else
                    upper |= 1 << (c - 'A');

                if (lower == upper && maxlength < j - i + 1) {
                    maxlength = j - i + 1;
                    strindex = i;
                }
            }

        }
        return strindex == -1 ? "" : s.substr(strindex, maxlength);


    }
```

Saved

Ln 1, Col 1

Testcase | Test Result

Description | Accepted × | Editorial | Solutions | Submissions

All Submissions

Accepted 130 / 130 testcases passed

Anirudh Singh submitted at Mar 18, 2025 21:27

Editorial | Solution

Runtime

28 ms | Beats 99.42%

Analyze Complexity

Memory

18.82 MB | Beats 36.76%

Code

C++ | Auto

```cpp
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        if (matrix.empty() || matrix[0].empty()) return false;

    int row = 0, col = matrix[0].size() - 1;

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

    return false;

    }
};
```

Saved

Ln 15, Col 1

Description | Accepted × | Editorial | Solutions | Submissions

All Submissions

Accepted 600 / 600 testcases passed

Anirudh Singh submitted at Mar 18, 2025 21:29

Editorial | Solution

Runtime

2 ms | Beats **44.50%**

Analyze Complexity

Memory

7.70 MB | Beats **87.34%**

Code

C++ | Auto

```cpp
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;

    }
};
```

Saved

Ln 9, Col 19

Description | Accepted × | Editorial | Solutions | Submissions

All Submissions

Accepted 598 / 598 testcases passed

Anirudh Singh submitted at Mar 18, 2025 21:30

Editorial | Solution

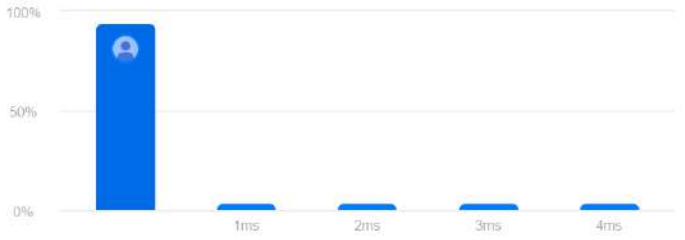
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

8.14 MB | Beats 80.26%



Code

C++ | Auto

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

    }
};
```

Saved

Ln 9, Col 18

Description | Accepted × | Editorial | Solutions | Submissions

All Submissions

Accepted 210 / 210 testcases passed

Anirudh Singh submitted at Mar 18, 2025 21:30

Editorial | Solution

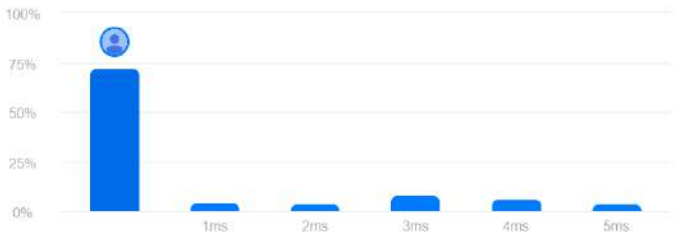
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

71.60 MB | Beats 97.82%



Code

C++ | Auto

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

        for (int i = 1; i < nums.size(); i++) {
            currentsum = max(nums[i], currentsum + nums[i]);
            maxsum = max(maxsum, currentsum);
        }

        return maxsum;
    }
};
```

Saved

Ln 1, Col 1

Description | Accepted × | Editorial | Solutions | Submissions

← All Submissions

Accepted 38 / 38 testcases passed

Anirudh Singh submitted at Mar 18, 2025 21:31

Editorial | Solution

Runtime

1 ms | Beats 52.28%

Analyze Complexity

Memory

9.98 MB | Beats 75.64%

Code

C++ | Auto

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

    }
};
```

Saved | Upgrade to Cloud Saving

Ln 15, Col 16

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 44 / 44 testcases passed

Anirudh Singh submitted at Mar 18, 2025 21:32

Editorial | Solution

Runtime

14 ms | Beats 69.41%

Analyze Complexity

Memory

27.77 MB | Beats 70.16%

Code

C++ | Auto

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

    for (auto& b : buildings) {
        points.emplace_back(b[0], -b[2]); // Start of building, height as negative
        points.emplace_back(b[1], b[2]);  // End of building, height as positive
    }

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

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

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

Saved

Ln 31, Col 19

Description | Accepted × | Editorial | Solutions | Submissions

All Submissions

Accepted 140 / 140 testcases passed

Anirudh Singh submitted at Mar 18, 2025 21:35

Editorial | Solution

Runtime

551 ms | Beats 26.11%

Analyze Complexity

Memory

240.22 MB | Beats 59.79%

Code

C++ | Auto

```cpp
class Solution {
public:
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);

        int 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++]);
```

Saved

Ln 33, Col 3

All Submissions

Accepted 84 / 84 testcases passed

Anirudh Singh submitted at Mar 18, 2025 21:41

Solution

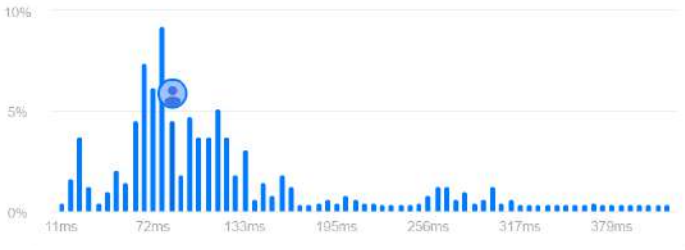
Runtime

85 ms | Beats 59.26%

Analyze Complexity

Memory

63.27 MB | Beats 71.40%



C++ | Auto

```cpp
class SegmentTree {
    vector<int> tree;
    int size;

public:
    SegmentTree(int n) {
        size = n;
        tree.resize(4 * n, 0);
    }

    void update(int index, int value, int node = 1, int left = 0, int right = -1) {
        if (right == -1) right = size - 1;
        if (left == right) {
            tree[node] = value;
            return;
        }

        int mid = left + (right - left) / 2;
        if (index <= mid) update(index, value, 2 * node, left, mid);
        else update(index, value, 2 * node + 1, mid + 1, right);

        tree[node] = max(tree[2 * node], tree[2 * node + 1]);
    }
```

Saved

Ln 51, Col 3