

# ASSIGNMENT 4

Problem List < > < Run Submit < > Premium

Description Accepted Solutions Editorial Submissions

All Submissions

Accepted 73 / 73 testcases passed

Shatakshi Sharma submitted at Mar 18, 2025 15:11

Solution

Runtime 8 ms | Beats 41.65%

Memory 14.22 MB | Beats 48.82%

Analyze Complexity

Code C++

```
1 class Solution {
2 public:
3     string longestNiceSubstring(string s) {
4         if (s.size() < 2)
5             return "";
6         unordered_set<char> st;
7         for (auto ch : s) {
8             st.insert(ch);
9         }
10        for (int i = 0; i < s.size(); i++) {
11            if (st.count(toupper(s[i])) && st.count(tolower(s[i]))) {
12                continue;
13            }
14            string prev = longestNiceSubstring(s.substr(0, i));
15            string next = longestNiceSubstring(s.substr(i + 1));
16            return prev.size() >= next.size() ? prev : next;
17        }
18        return s;
19    };
20 }
```

Testcase Test Result

Problem List < > < Run Submit < > Premium

Description Solutions Accepted Editorial Submissions

All Submissions

Accepted 600 / 600 testcases passed

Shatakshi Sharma submitted at Mar 18, 2025 15:14

Editorial Solution

Runtime 0 ms | Beats 100.00%

Memory 7.79 MB | Beats 63.34%

Analyze Complexity

Code C++

```
1 class Solution {
2 public:
3     uint32_t reverseBits(uint32_t n) {
4         string bits = "";
5
6         for(int i = 0; i < 32; i++){
7             if(n & (1 << i)) bits.push_back('1');
8             else bits.push_back('0');
9         }
10
11        uint32_t ans = 0;
12        for(int i = 31, j = 0; i >= 0; i--){
13            if(bits[j++] == '1') ans = ans | (1 << i);
14        }
15        return ans;
16    };
17 }
```

Problem List < > < Run Submit < > Premium

Description Solutions Accepted Editorial Submissions

All Submissions

Accepted 598 / 598 testcases passed

Shatakshi Sharma submitted at Mar 18, 2025 15:15

Editorial Solution

Runtime 0 ms | Beats 100.00%

Memory 8.32 MB | Beats 12.20%

Analyze Complexity

Code C++

```
1 class Solution {
2 public:
3     int hammingWeight(uint32_t n) {
4         int res = 0;
5         for (int i = 0; i < 32; i++) {
6             if ((n >> i) & 1) {
7                 res += 1;
8             }
9         }
10        return res;
11    };
12 }
```

Testcase Test Result

Problem List < > ✕

Run Submit

Description Accepted Solutions Editorial Submissions

All Submissions

Accepted 210 / 210 testcases passed

Shatakshi Sharma submitted at Mar 18, 2025 15:16

Editorial Solution

Runtime 0 ms | Beats 100.00%

Memory 71.84 MB | Beats 18.41%

Analyze Complexity

Code C++

```
1 class Solution {
2 public:
3     int maxSubArray(vector<int>& nums) {
4         int maxSum = INT_MIN;
5         int currentSum = 0;
6         for (int num : nums) {
7             currentSum = max(num, currentSum + num);
8             maxSum = max(maxSum, currentSum);
9         }
10        return maxSum;
11    }
12};
```

Problem List < > ✕

Run Submit

Description Accepted Solutions Editorial Submissions

All Submissions

Accepted 130 / 130 testcases passed

Shatakshi Sharma submitted at Mar 18, 2025 15:17

Editorial Solution

Runtime 93 ms | Beats 18.95%

Memory 18.80 MB | Beats 36.76%

Analyze Complexity

Code C++

```
1 class Solution {
2 public:
3     bool searchMatrix(vector<vector<int>>& matrix, int target) {
4         for (int i = 0; i < matrix.size(); i++) {
5             int start = 0;
6             int end = matrix[i].size() - 1;
7
8             while (start <= end) {
9
10                int mid = start + (end - start) / 2;
11                if (matrix[i][mid] == target) {
12                    return true;
13                }
14                else if (matrix[i][mid] < target) {
15                    start = mid + 1;
16                }
17                else {
18                    end = mid - 1;
19                }
20            }
21        }
22        return false;
23    }
```

Problem List < > ✕

Run Submit

Description Solutions Accepted Editorial Submissions

All Submissions

Accepted 57 / 57 testcases passed

Shatakshi Sharma submitted at Mar 18, 2025 15:18

Solution

Runtime 0 ms | Beats 100.00%

Memory 15.31 MB | Beats 15.09%

Analyze Complexity

Code C++

```
1 class Solution {
2     const int base = 1337;
3     int powmod(int a, int k) {
4         a %= base;
5         int result = 1;
6         for (int i = 0; i < k; ++i) {
7             result = (result * a) % base;
8         }
9         return result;
10    }
11 public:
12    int superPow(int a, vector<int>& b) {
13        if (b.empty()) return 1;
14        int last_digit = b.back();
15        b.pop_back();
16        return powmod(superPow(a, b), 10) * powmod(a, last_digit) % base;
17    }
18};
```

Problem List < > ✕

Run Submit

Description Solutions Accepted x Editorial Submissions

← All Submissions

Accepted 38 / 38 testcases passed

Shatakshi Sharma submitted at Mar 18, 2025 15:20

Editorial Solution

Runtime 0 ms | Beats 100.00%

Memory 9.50 MB | Beats 89.46%

Analyze Complexity

Code C++

```
1 class Solution {
2 public:
3     int partition(vector<int> &v, int start, int end, int mask)
4     {
5         int j = start;
6         for(int i = start; i <= end; i++)
7         {
8             if((v[i] & mask) != 0)
9             {
10                 swap(v[i], v[j]);
11                 j++;
12             }
13         }
14         return j;
15     }
16
17     void sort(vector<int> &v, int start, int end, int mask)
18     {
19         if(start >= end) return;
20         int mid = partition(v, start, end, mask);
21         sort(v, start, mid - 1, mask << 1);
22         sort(v, mid, end, mask << 1);
23     }
24 }
```

Saved Ln 31, Col 3

Testcase Test Result

Problem List < > ✕

Run Submit

Description Solutions Accepted x Editorial Submissions

← All Submissions

Accepted 44 / 44 testcases passed

Shatakshi Sharma submitted at Mar 18, 2025 15:21

Editorial Solution

Runtime 11 ms | Beats 86.88%

Memory 26.48 MB | Beats 96.21%

Analyze Complexity

Code C++

```
1 class Solution {
2 public:
3     vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
4         int edge_idx = 0;
5         vector<pair<int, int>> edges;
6         priority_queue<pair<int, int>> pq;
7         vector<vector<int>> skyline;
8
9         for (int i = 0; i < buildings.size(); ++i) {
10             const auto &b = buildings[i];
11             edges.emplace_back(b[0], 1);
12             edges.emplace_back(b[1], 1);
13         }
14
15         std::sort(edges.begin(), edges.end());
16
17         while (edge_idx < edges.size()) {
18             int curr_height;
19             const auto &[curr_x, _] = edges[edge_idx];
20             while (edge_idx < edges.size() &&
21                  curr_x == edges[edge_idx].first) {
22                 const auto &[_, building_idx] = edges[edge_idx];
23                 const auto &b = buildings[building_idx];
24             }
25         }
26     }
27 }
```

Saved Ln 36, Col 3

Problem List < > ✕

Run Submit

Description Solutions Accepted x Editorial Submissions

← All Submissions

Accepted 140 / 140 testcases passed

Shatakshi Sharma submitted at Mar 18, 2025 15:23

Editorial Solution

Runtime 74 ms | Beats 96.45%

Memory 52.62 MB | Beats 94.62%

Analyze Complexity

Code C++

```
1 class Solution {
2 public:
3     void update(vector<int>& BIT, int index, int val)
4     {
5         while (index > 0) {
6             BIT[index] += val;
7             index -= index & (-index);
8         }
9     }
10
11     int query(vector<int>& BIT, int index)
12     {
13         int sum = 0;
14         while (index < BIT.size()) {
15             sum += BIT[index];
16             index += index & (-index);
17         }
18         return sum;
19     }
20
21     int reversePairs(vector<int>& nums)
22     {
23         int n = nums.size();
24         vector<int> nums_copy(nums);
25     }
26 }
```

Saved Ln 34, Col 2

Testcase Test Result

Problem List

RunSubmit

8817Premium

DescriptionSolutionsAcceptedEditorialSubmissions

All Submissions

Accepted 84 / 84 testcases passed  
Shatakshi Sharma submitted at Mar 18, 2025 15:25

Runtime  
111 ms | Beats 44.03%  
Analyze Complexity

Memory  
143.63 MB | Beats 26.95%

11ms 72ms 133ms 195ms 256ms 317ms 379ms

Code C++

Code

C++Auto

```
1 class MaxSegmentTree {
2 public:
3     int n;
4     vector<int> tree;
5     MaxSegmentTree(int n_) : n(n_) {
6         int size = (int)(ceil(log2(n)));
7         size = (2 * pow(2, size)) - 1;
8         tree = vector<int>(size);
9     }
10
11     int max_value() { return tree[0]; }
12
13     int query(int l, int r) { return query_util(0, l, r, 0, n - 1); }
14
15     int query_util(int i, int ql, int qr, int l, int r) {
16         if (l >= ql && r <= qr) return tree[i];
17         if (l > qr || r < ql) return INT_MIN;
18
19         int m = (l + r) / 2;
20         return max(query_util(2 * i + 1, ql, qr, l, m), query_util(2 * i + 2, ql, qr,
21 m + 1, r));
22     }
23 }
```

SavedLn 50, Col 3

TestcaseTest Result