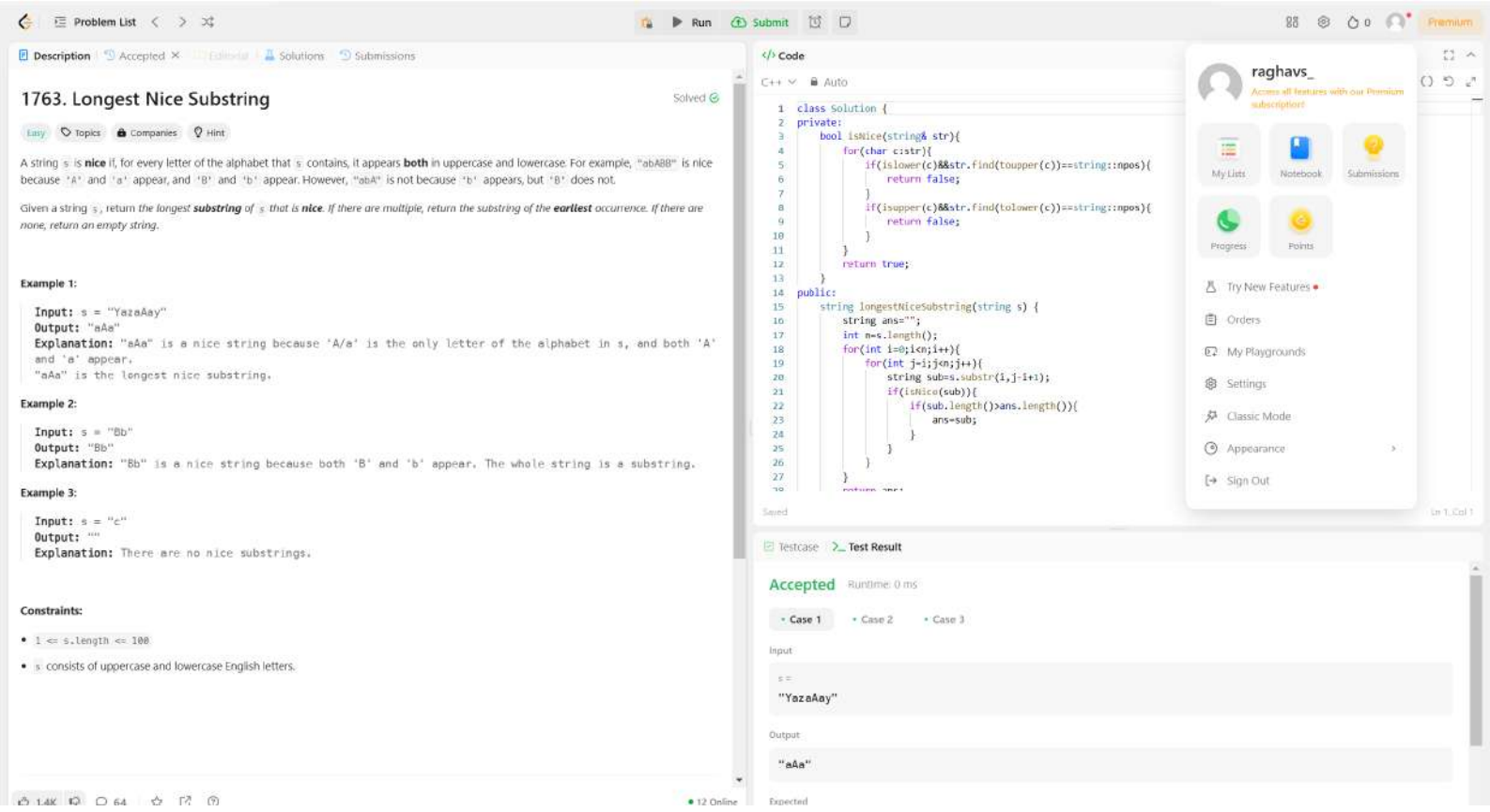
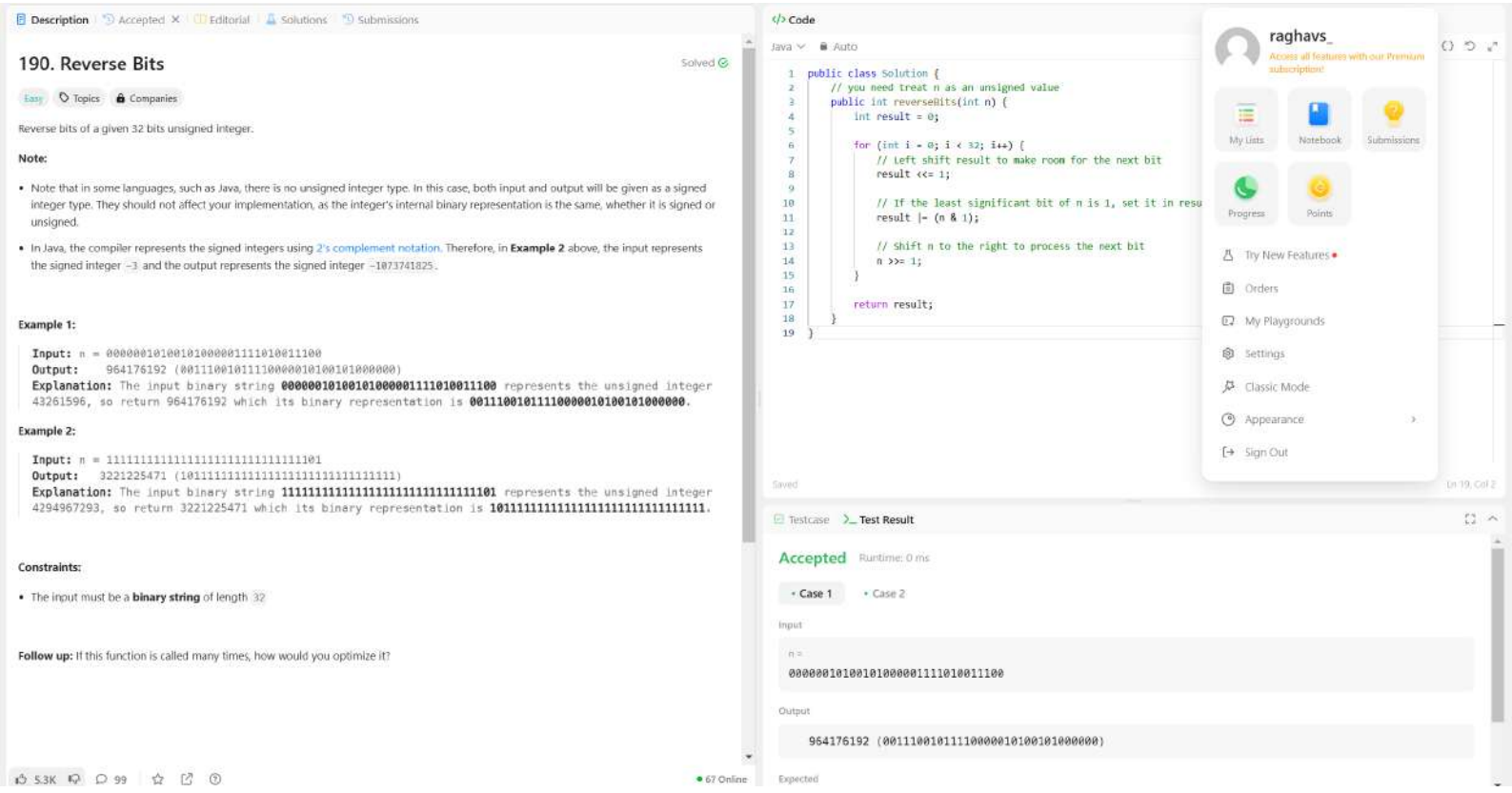
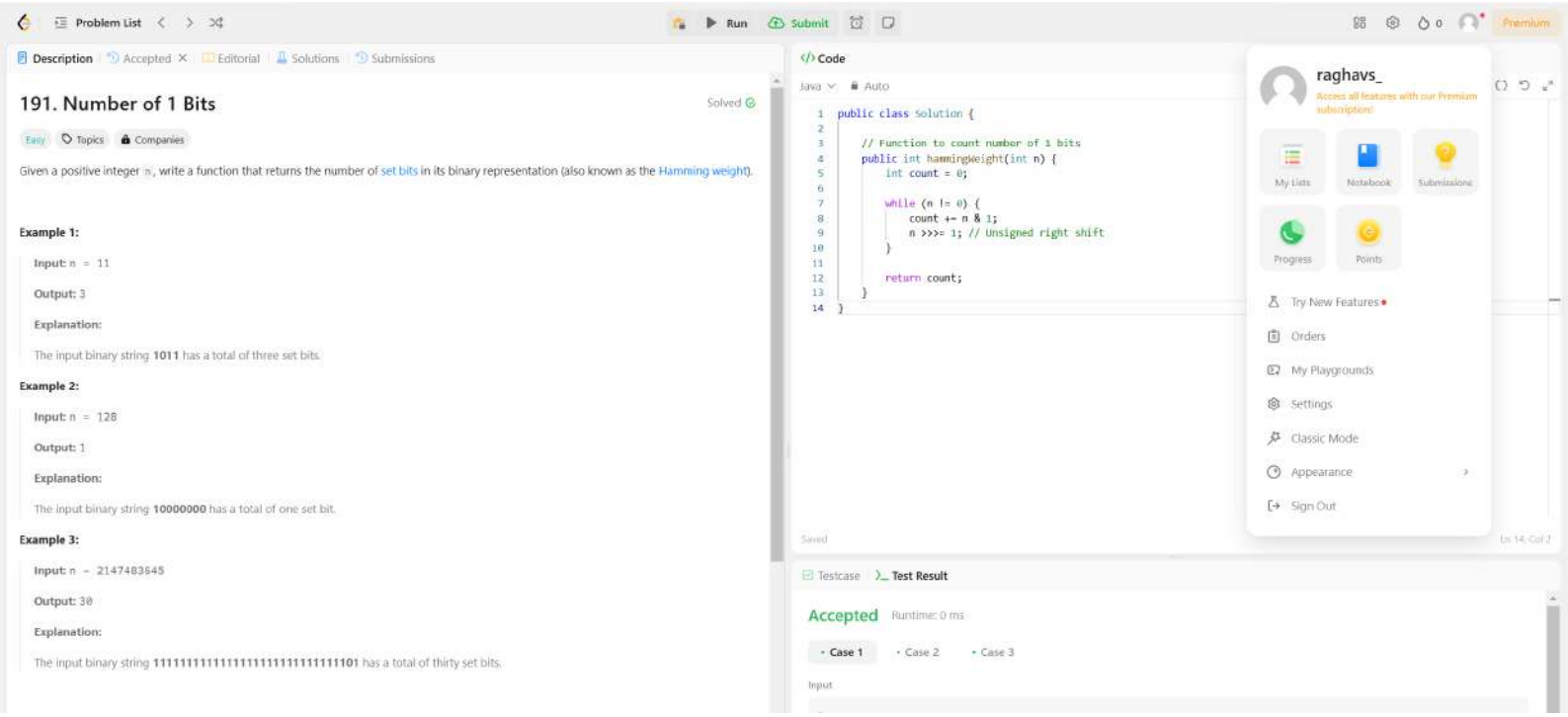
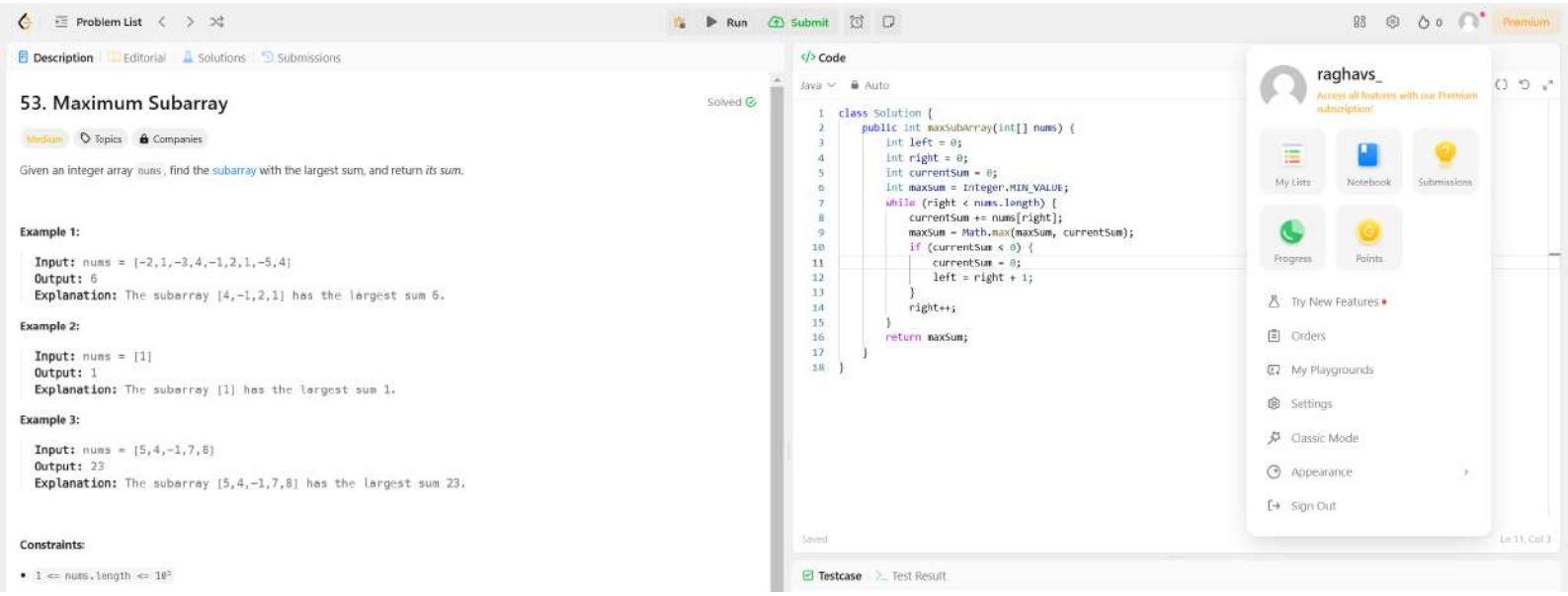
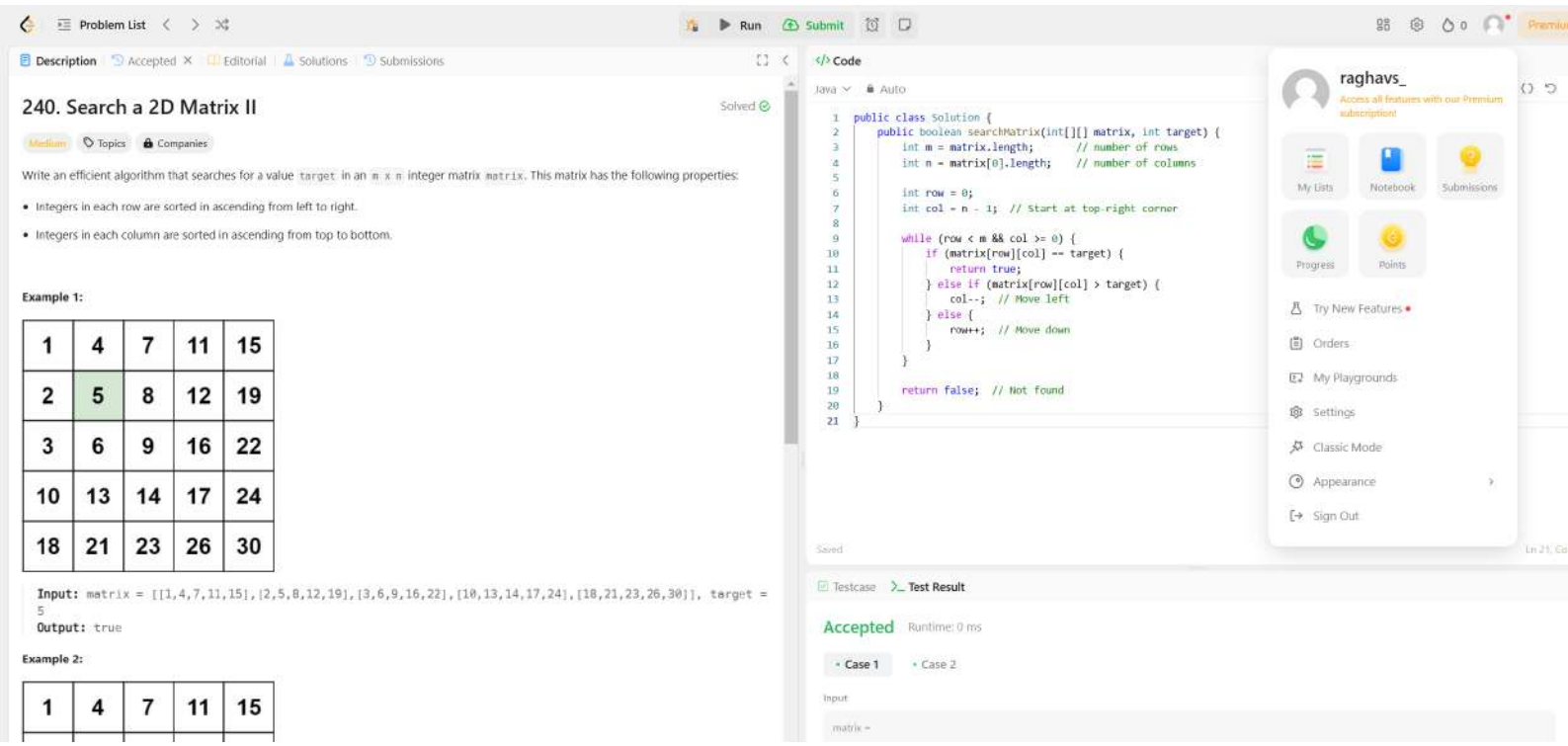
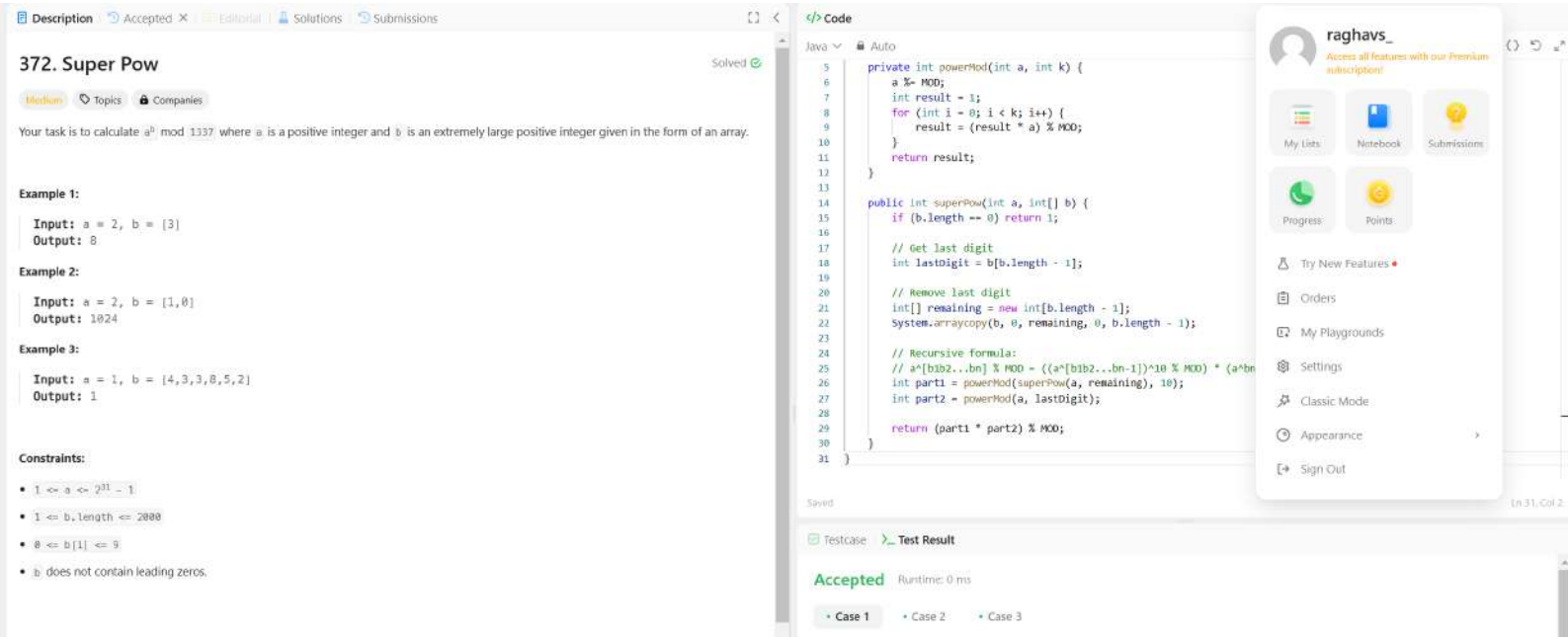

Problem List
Run
Submit
Premium
Description
Accepted
Solutions
Submissions
1763. Longest Nice Substring
Solved
Easy
Topics
Companies
Hint
A string s is nice if, for every letter of the alphabet that s contains, it appears both in uppercase and lowercase. For example, "abABB" is nice because 'A' and 'a' appear, and 'B' and 'b' appear. However, "abA" is not because 'b' appears, but 'B' does not.
Given a string s, return the longest substring of s that is nice. If there are multiple, return the substring of the earliest occurrence. If there are none, return an empty string.
Example 1:
Input: s = "YazaAay"
Output: "aAa"
Explanation: "aAa" is a nice string because 'A/a' is the only letter of the alphabet in s, and both 'A' and 'a' appear.
"aAa" is the longest nice substring.
Example 2:
Input: s = "Bb"
Output: "Bb"
Explanation: "Bb" is a nice string because both 'B' and 'b' appear. The whole string is a substring.
Example 3:
Input: s = "c"
Output: ""
Explanation: There are no nice substrings.
Constraints:
1 <= s.length <= 100
s consists of uppercase and lowercase English letters.
1.4K
64
12 Online
Code
C++
Auto
class Solution {
private:
bool isNice(string& str){
for(char c:str){
if(islower(c)&&str.find(toupper(c))==string::npos){
return false;
}
if(isupper(c)&&str.find(tolower(c))==string::npos){
return false;
}
}
return true;
}
public:
string longestNiceSubstring(string s) {
string ans="";
int n=s.length();
for(int i=0;i<n;i++){
for(int j=i;j<n;j++){
string sub=s.substr(i,j-i+1);
if(isNice(sub)){
if(sub.length()>ans.length()){
ans=sub;
}
}
}
}
Saved
Ln 1, Col 1
raghavs_
Access all features with our Premium subscription!
My Lists
Notebook
Submissions
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Classic Mode
Appearance
Sign Out
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
s =
"YazaAay"
Output
"aAa"
Expected

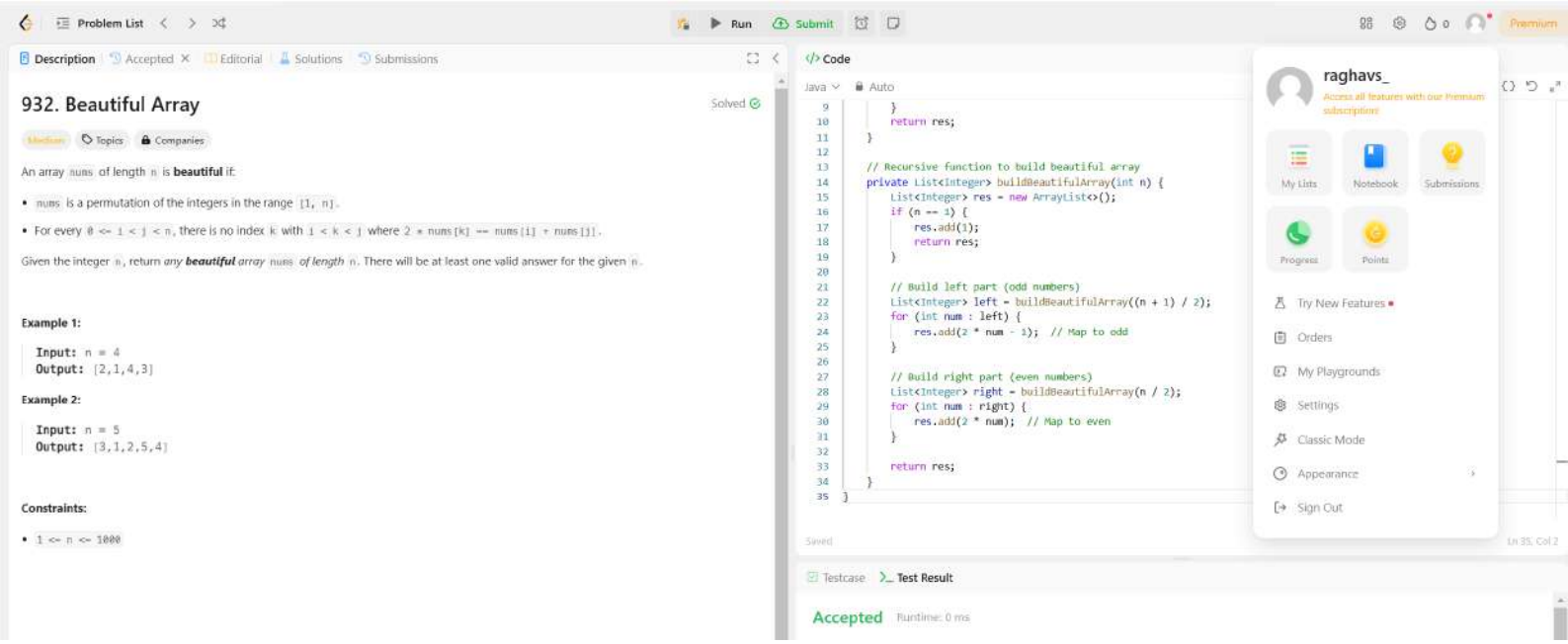

Description | Accepted | Editorial | Solutions | Submissions
190. Reverse Bits
Solved
Easy
Topics
Companies
Reverse bits of a given 32 bits unsigned integer.
Note:
Note that in some languages, such as Java, there is no unsigned integer type. In this case, both input and output will be given as a signed integer type. They should not affect your implementation, as the integer's internal binary representation is the same, whether it is signed or unsigned.
In Java, the compiler represents the signed integers using 2's complement notation. Therefore, in Example 2 above, the input represents the signed integer -3 and the output represents the signed integer -1073741825.
Example 1:
Input: n = 00000010100101000001111010011100
Output: 964176192 (00111001011110000010100101000000)
Explanation: The input binary string 00000010100101000001111010011100 represents the unsigned integer 43261596, so return 964176192 which its binary representation is 00111001011110000010100101000000.
Example 2:
Input: n = 11111111111111111111111111111101
Output: 3221225471 (10111111111111111111111111111111)
Explanation: The input binary string 11111111111111111111111111111101 represents the unsigned integer 4294967293, so return 3221225471 which its binary representation is 10111111111111111111111111111111.
Constraints:
The input must be a binary string of length 32
Follow up: If this function is called many times, how would you optimize it?
5.3K
99
67 Online
Code
Java
Auto
1 public class Solution {
2 // you need treat n as an unsigned value
3 public int reverseBits(int n) {
4 int result = 0;
5
6 for (int i = 0; i < 32; i++) {
7 // Left shift result to make room for the next bit
8 result <<= 1;
9
10 // If the least significant bit of n is 1, set it in resu
11 result |= (n & 1);
12
13 // Shift n to the right to process the next bit
14 n >>= 1;
15 }
16
17 return result;
18 }
19 }
raghavs_
Access all features with our Premium subscription!
My Lists
Notebook
Submissions
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Classic Mode
Appearance
Sign Out
Saved
Ln 19, Col 2
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
n =
00000010100101000001111010011100
Output
964176192 (00111001011110000010100101000000)
Expected


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
191. Number of 1 Bits
Solved
Easy
Topics
Companies
Given a positive integer n, write a function that returns the number of set bits in its binary representation (also known as the Hamming weight).
Example 1:
Input: n = 11
Output: 3
Explanation:
The input binary string 1011 has a total of three set bits.
Example 2:
Input: n = 128
Output: 1
Explanation:
The input binary string 10000000 has a total of one set bit.
Example 3:
Input: n = 2147483645
Output: 30
Explanation:
The input binary string 11111111111111111111111111111101 has a total of thirty set bits.
Code
Java
Auto
public class Solution {
// Function to count number of 1 bits
public int hammingWeight(int n) {
int count = 0;
while (n != 0) {
count += n & 1;
n >>>= 1; // Unsigned right shift
}
return count;
}
}
raghavs_
Access all features with our Premium subscription!
My Lists
Notebook
Submissions
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Classic Mode
Appearance
Sign Out
Saved
Ln 14, Col 2
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input


Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
53. Maximum Subarray
Solved
Medium
Topics
Companies
Given an integer array nums, find the subarray with the largest sum, and return its sum.
Example 1:
Input: nums = [-2,1,-3,4,-1,2,1,-5,4]
Output: 6
Explanation: The subarray [4,-1,2,1] has the largest sum 6.
Example 2:
Input: nums = [1]
Output: 1
Explanation: The subarray [1] has the largest sum 1.
Example 3:
Input: nums = [5,4,-1,7,8]
Output: 23
Explanation: The subarray [5,4,-1,7,8] has the largest sum 23.
Constraints:
1 <= nums.length <= 10^5
Code
Java
Auto
class Solution {
public int maxSubArray(int[] nums) {
int left = 0;
int right = 0;
int currentSum = 0;
int maxSum = Integer.MIN_VALUE;
while (right < nums.length) {
currentSum += nums[right];
maxSum = Math.max(maxSum, currentSum);
if (currentSum < 0) {
currentSum = 0;
left = right + 1;
}
right++;
}
return maxSum;
}
}
raghavs_
Access all features with our Premium subscription!
My Lists
Notebook
Submissions
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Classic Mode
Appearance
Sign Out
Saved
Ln 11, Col 3
Testcase
Test Result


Problem List
Run
Submit
Premiu
Description
Accepted
Editorial
Solutions
Submissions
240. Search a 2D Matrix II
Solved
Medium
Topics
Companies
Write an efficient algorithm that searches for a value target in an m x n integer matrix matrix. This matrix has the following properties:
Integers in each row are sorted in ascending from left to right.
Integers in each column are sorted in ascending from top to bottom.
Example 1:
1 4 7 11 15
2 5 8 12 19
3 6 9 16 22
10 13 14 17 24
18 21 23 26 30
Input: matrix = [[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]], target = 5
Output: true
Example 2:
1 4 7 11 15
Code
Java
Auto
public class Solution {
public boolean searchMatrix(int[][] matrix, int target) {
int m = matrix.length; // number of rows
int n = matrix[0].length; // number of columns
int row = 0;
int col = n - 1; // Start at top-right corner
while (row < m && col >= 0) {
if (matrix[row][col] == target) {
return true;
} else if (matrix[row][col] > target) {
col--; // Move left
} else {
row++; // Move down
}
}
return false; // Not found
}
}
Saved
raghavs_
Access all features with our Premium subscription!
My Lists
Notebook
Submissions
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Classic Mode
Appearance
Sign Out
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
matrix =


Description
Accepted
Editorial
Solutions
Submissions
372. Super Pow
Solved
Medium
Topics
Companies
Your task is to calculate a^b mod 1337 where a is a positive integer and b is an extremely large positive integer given in the form of an array.
Example 1:
Input: a = 2, b = [3]
Output: 8
Example 2:
Input: a = 2, b = [1,0]
Output: 1024
Example 3:
Input: a = 1, b = [4,3,3,8,5,2]
Output: 1
Constraints:
1 <= a <= 2^31 - 1
1 <= b.length <= 2000
0 <= b[i] <= 9
b does not contain leading zeros.
Code
Java
Auto
private int powerMod(int a, int k) {
a %= MOD;
int result = 1;
for (int i = 0; i < k; i++) {
result = (result * a) % MOD;
}
return result;
}
public int superPow(int a, int[] b) {
if (b.length == 0) return 1;
// Get last digit
int lastDigit = b[b.length - 1];
// Remove last digit
int[] remaining = new int[b.length - 1];
System.arraycopy(b, 0, remaining, 0, b.length - 1);
// Recursive formula:
// a^[b1b2...bn] % MOD = ((a^[b1b2...bn-1])^10 % MOD) * (a^bn
int part1 = powerMod(superPow(a, remaining), 10);
int part2 = powerMod(a, lastDigit);
return (part1 * part2) % MOD;
}
}
raghavs_
Access all features with our Premium subscription!
My Lists
Notebook
Submissions
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Classic Mode
Appearance
Sign Out
Saved
Ln 31, Col 2
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
932. Beautiful Array
Solved
Medium
Topics
Companies
An array nums of length n is beautiful if:
nums is a permutation of the integers in the range [1, n].
For every 0 <= i < j < n, there is no index k with i < k < j where 2 * nums[k] == nums[i] + nums[j].
Given the integer n, return any beautiful array nums of length n. There will be at least one valid answer for the given n.
Example 1:
Input: n = 4
Output: [2,1,4,3]
Example 2:
Input: n = 5
Output: [3,1,2,5,4]
Constraints:
1 <= n <= 1000
Code
Java
Auto
}
return res;
}
// Recursive function to build beautiful array
private List<Integer> buildBeautifulArray(int n) {
List<Integer> res = new ArrayList<>();
if (n == 1) {
res.add(1);
return res;
}
// Build left part (odd numbers)
List<Integer> left = buildBeautifulArray((n + 1) / 2);
for (int num : left) {
res.add(2 * num - 1); // Map to odd
}
// Build right part (even numbers)
List<Integer> right = buildBeautifulArray(n / 2);
for (int num : right) {
res.add(2 * num); // Map to even
}
return res;
}
}
raghavs_
Access all features with our Premium subscription!
My Lists
Notebook
Submissions
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Classic Mode
Appearance
Sign Out
Saved
Ln 35, Col 2
Testcase
Test Result
Accepted
Runtime: 0 ms

Problem List

Run Submit

Premium

Description | Accepted | Editorial | Solutions | Submissions

218. The Skyline Problem

Solved

Hard | Topics | Companies

A city's **skyline** is the outer contour of the silhouette formed by all the buildings in that city when viewed from a distance. Given the locations and heights of all the buildings, return *the* ***skyline*** *formed by these buildings collectively.*

The geometric information of each building is given in the array `buildings` where `buildings[i] = [left_i, right_i, height_i]`:

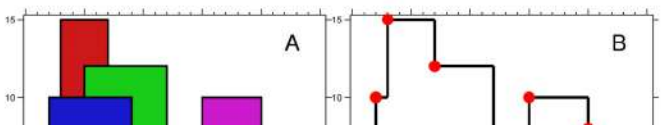
- `left_i` is the x coordinate of the left edge of the i^{th} building.
- `right_i` is the x coordinate of the right edge of the i^{th} building.
- `height_i` is the height of the i^{th} building.

You may assume all buildings are perfect rectangles grounded on an absolutely flat surface at height `0`.

The **skyline** should be represented as a list of "key points" **sorted by their x-coordinate** in the form `[[x1,y1],[x2,y2],...]`. Each key point is the left endpoint of some horizontal segment in the skyline except the last point in the list, which always has a y-coordinate `0` and is used to mark the skyline's termination where the rightmost building ends. Any ground between the leftmost and rightmost buildings should be part of the skyline's contour.

Note: There must be no consecutive horizontal lines of equal height in the output skyline. For instance, `[...,[2 3],[4 5],[7 5],[11 5],[12 7],...]` is not acceptable; the three lines of height 5 should be merged into one in the final output as such: `[...,[2 3],[4 5],[12 7],...]`

Example 1:



Code

Java | Auto

```java
        maxHeap.add(0); // Initial ground height

        int prevMax = 0;
        List<List<Integer>> result = new ArrayList<>();

        for (int[] e : events) {
            int x = e[0];
            int h = e[1];

            if (h < 0) {
                // Start of building
                maxHeap.add(-h);
            } else {
                // End of building
                maxHeap.remove(h);
            }

            int currMax = maxHeap.peek();
            if (currMax != prevMax) {
                result.add(Arrays.asList(x, currMax));
                prevMax = currMax;
            }
        }

        return result;
    }
}
```

Saved

Ln 48, Col 2

raghavs_

Access all features with our Premium subscription!

My Lists | Notebook | Submissions | Progress | Points

Try New Features

Orders

My Playgrounds

Settings

Classic Mode

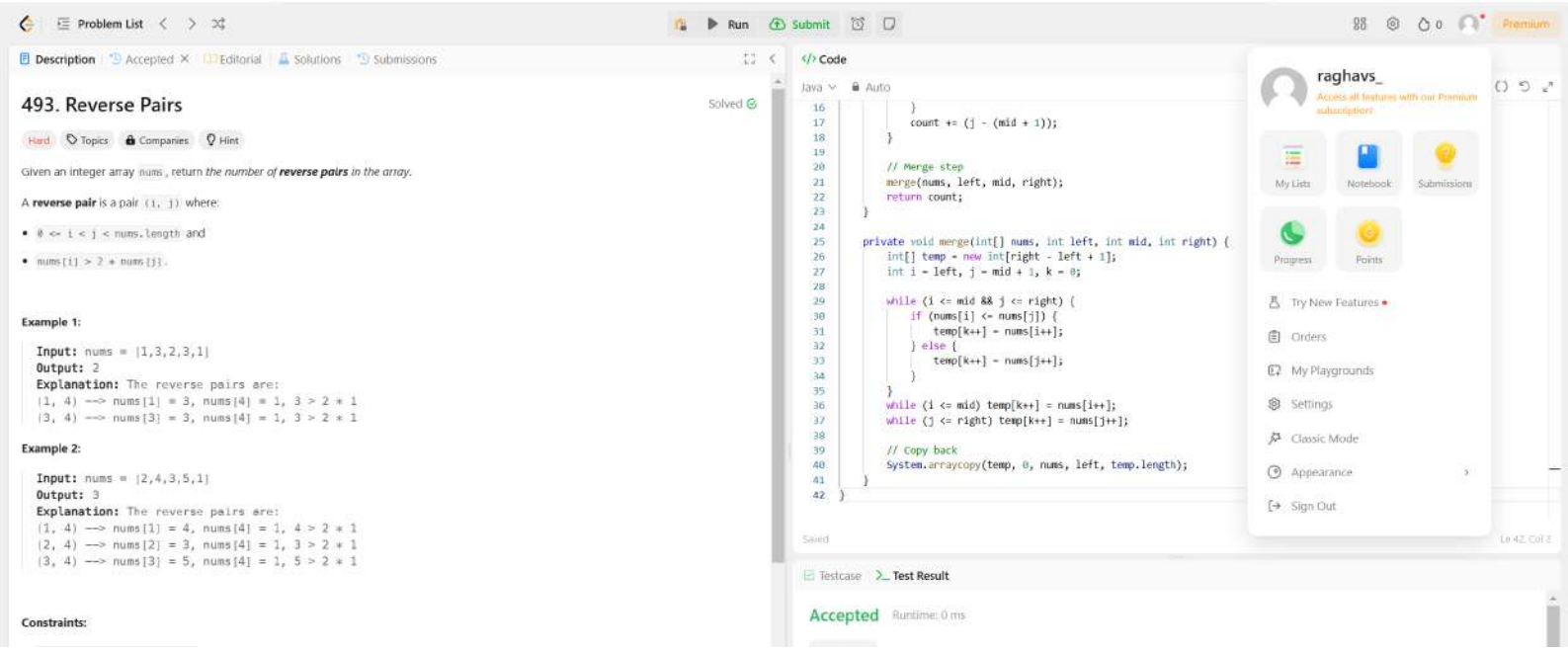
Appearance

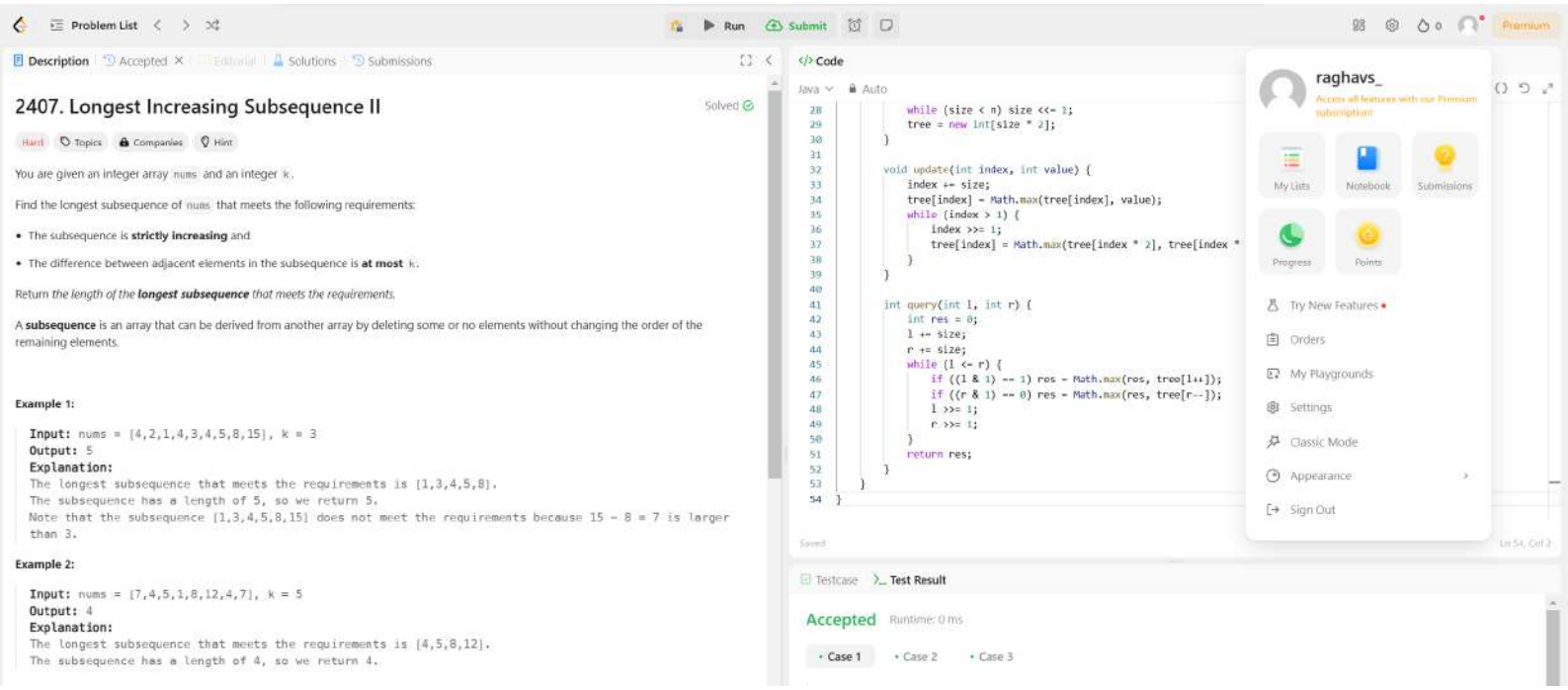
Sign Out

Testcase | Test Result

Accepted Runtime: 1 ms

Case 1 | Case 2


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
493. Reverse Pairs
Solved
Hard
Topics
Companies
Hint
Given an integer array nums, return the number of reverse pairs in the array.
A reverse pair is a pair (i, j) where:
0 <= i < j < nums.length and
nums[i] > 2 * nums[j].
Example 1:
Input: nums = [1,3,2,3,1]
Output: 2
Explanation: The reverse pairs are:
(1, 4) --> nums[1] = 3, nums[4] = 1, 3 > 2 * 1
(3, 4) --> nums[3] = 3, nums[4] = 1, 3 > 2 * 1
Example 2:
Input: nums = [2,4,3,5,1]
Output: 3
Explanation: The reverse pairs are:
(1, 4) --> nums[1] = 4, nums[4] = 1, 4 > 2 * 1
(2, 4) --> nums[2] = 3, nums[4] = 1, 3 > 2 * 1
(3, 4) --> nums[3] = 5, nums[4] = 1, 5 > 2 * 1
Constraints:
Code
Java
Auto
count += (j - (mid + 1));
// Merge step
merge(nums, left, mid, right);
return count;
private void merge(int[] nums, int left, int mid, int right) {
int[] temp = new int[right - left + 1];
int i = left, j = mid + 1, k = 0;
while (i <= mid && j <= right) {
if (nums[i] <= nums[j]) {
temp[k++] = nums[i++];
} else {
temp[k++] = nums[j++];
while (i <= mid) temp[k++] = nums[i++];
while (j <= right) temp[k++] = nums[j++];
// Copy back
System.arraycopy(temp, 0, nums, left, temp.length);
Saved
Ln 42, Col 2
raghavs_
Access all features with our Premium subscription!
My Lists
Notebook
Submissions
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Classic Mode
Appearance
Sign Out
Testcase
Test Result
Accepted
Runtime: 0 ms


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
2407. Longest Increasing Subsequence II
Solved
Hard
Topics
Companies
Hint
You are given an integer array nums and an integer k.
Find the longest subsequence of nums that meets the following requirements:
The subsequence is strictly increasing and
The difference between adjacent elements in the subsequence is at most k.
Return the length of the longest subsequence that meets the requirements.
A subsequence is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements.
Example 1:
Input: nums = [4,2,1,4,3,4,5,8,15], k = 3
Output: 5
Explanation:
The longest subsequence that meets the requirements is [1,3,4,5,8].
The subsequence has a length of 5, so we return 5.
Note that the subsequence [1,3,4,5,8,15] does not meet the requirements because 15 - 8 = 7 is larger than 3.
Example 2:
Input: nums = [7,4,5,1,8,12,4,7], k = 5
Output: 4
Explanation:
The longest subsequence that meets the requirements is [4,5,8,12].
The subsequence has a length of 4, so we return 4.
Code
Java
Auto
while (size < n) size <<= 1;
tree = new int[size * 2];
}
void update(int index, int value) {
index += size;
tree[index] = Math.max(tree[index], value);
while (index > 1) {
index >>= 1;
tree[index] = Math.max(tree[index * 2], tree[index *
}
}
int query(int l, int r) {
int res = 0;
l += size;
r += size;
while (l <= r) {
if ((l & 1) == 1) res = Math.max(res, tree[l++]);
if ((r & 1) == 0) res = Math.max(res, tree[r--]);
l >>= 1;
r >>= 1;
}
return res;
}
}
}
raghavs_
Access all features with our Premium subscription!
My Lists
Notebook
Submissions
Progress
Points
Try New Features
Orders
My Playgrounds
Settings
Classic Mode
Appearance
Sign Out
Saved
Ln 54, Col 2
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3