

Problem List

RunSubmit

Premium

DescriptionEditorialSolutionsAcceptedSubmissions

1763. Longest Nice Substring

Solved

EasyTopicsCompaniesHint

A string `s` is **nice** if, for every letter of the alphabet that `s` contains, it appears **both** in uppercase and lowercase. For example, `"abAB"` 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.

1.4K6025 Online

Code

C++Auto

```
1 class Solution
2 {
3 public:
4     string longestNiceSubstring(string s) {
5         if (s.size() < 2) return "";
6         unordered_set<char> st(begin(s), end(s));
7         for (int i = 0; i < s.size(); i++) {
8             if (st.find((char) toupper(s[i])) == end(st) || st.find((char) tolower(s[i])) == end(st))
9                 continue;
10            string s1 = longestNiceSubstring(s.substr(0, i));
11            string s2 = longestNiceSubstring(s.substr(i + 1));
12            return s1.size() >= s2.size() ? s1 : s2;
13        }
14        return s;
15    }
16 };
```

SavedLn 16, Col 3

TestcaseTest Result

AcceptedRuntime: 0 ms

DescriptionEditorialSolutionsAcceptedSubmissions

190. Reverse Bits

EasyTopicsCompanies

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:

Solved

5.3K9894 Online

Code

C++Auto

```
1 class Solution {
2 public:
3     uint32_t reverseBits(uint32_t n) {
4
5         n = ((n & 0xffff0000) >> 16) | ((n & 0x0000ffff) << 16);
6         n = ((n & 0xff00ff00) >> 8) | ((n & 0x00ff00ff) << 8);
7         n = ((n & 0xf0f0f0f0) >> 4) | ((n & 0x0f0f0f0f) << 4);
8         n = ((n & 0xcccccccc) >> 2) | ((n & 0x33333333) << 2);
9         n = ((n & 0xaaaaaaaa) >> 1) | ((n & 0x55555555) << 1);
10
11         return n;
12     }
13 }
14
```

SavedLn 14, Col 3

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2

Input

n =

Problem List

Run

Submit

Premium

Description

Editorial

Solutions

Accepted

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.

6.7K17570 Online

Code

C++

Auto

```
1 class Solution {
2 public:
3     int hammingWeight(int n) {
4         int count = 0;
5         for(int i = 31; i >= 0; i--){
6             if(((n >> i) & 1) == 1)
7                 count++;
8         }
9         return count;
10    }
11};
```

SavedLn 11, Col 3

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2Case 3

Input

n = 11

Output

Problem List

Run

Submit

0

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.

35.2K 327 540 Online

Code

C++

Auto

```
1 class Solution {
2 public:
3     int maxarray(vector<int>&v,int l,int r){
4         if(l>r){
5             return INT_MIN;
6         }
7         int mid=(l+r)/2,leftsum=0,rightsum=0;
8         for(int i=mid-1;cursum=0;i>=1;i--){
9             cursum+=v[i];
10            leftsum=max(cursum,leftsum);
11        }
12        for(int i=mid+1;cursum=0;i<=r;i++){
13            cursum+=v[i];
14            rightsum=max(cursum,rightsum);
15        }
16        return max(maxarray(v,mid+1,r),maxarray(v,l,mid-1),v[mid]+leftsum+rightsum);
17    }
18    int maxSubArray(vector<int>& nums) {
19        return maxarray(nums,0,nums.size()-1);
20    }
21 }
```

Saved

Ln 1, Col 1

Testcase

Test Result

Accepted

Runtime: 0 ms

Case 1

Case 2

Case 3

Problem List

Run

Submit

0

Premium

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

12.3K

82

62 Online

Code

C++

Auto

```
1 class Solution {
2 public:
3     bool searchMatrix(vector<vector<int>>& matrix, int target) {
4         int n = matrix.size(), m = matrix[0].size();
5         int row = 0, col = m - 1;
6
7         while (row < n && col >= 0) {
8             if (matrix[row][col] == target) return true;
9             else if (matrix[row][col] < target) row++;
10            else col--;
11        }
12        return false;
13    }
14};
```

Saved

Ln 14, Col 3

Testcase

Test Result

Accepted

Runtime: 3 ms

Case 1

Case 2

Input

matrix =

Problem List

Run

Submit

Premium

Description

Editorial

Solutions

Submissions

372. Super Pow

Medium

Topics

Companies

Your task is to calculate $a^b \bmod 1337$ 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:

992 23 8 Online

Code

C++

Auto

```
1 class Solution {
2     const int base = 1337;
3     int powmod(int a, int k)
4     {
5         a %= base;
6         int result = 1;
7         for (int i = 0; i < k; ++i)
8             result = (result * a) % base;
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 };
```

Saved

Ln 3, Col 30

Testcase

Test Result

Accepted

Runtime: 0 ms

Case 1

Case 2

Case 3

Input