Experiment 6

Student Name: Abhinav singh minhas	**UID: 20BCS9633**
Branch: BE-CSE	**Section/Group: 608-A**
Semester: 6TH	**Date of Performance:5/04/2023**
Subject Name: C.C LAB	**Subject Code: 20CSP-351**

1. Aim

To implement the concept of Graph.

2. Objective

1.The objective is to build problem solving capability and to learn the basic concepts of data structures.

2.The implementation of Same Tree which shows and brushes up the concept of Graphs .

3.The implementation of Graphs.

3. Algorithm

1. Assign two String s and t.
2. Take XOR operation of every character.
3. All the n character of s "abc" is similar to n character of t "cab". So, they will cancel each other.
4. And we left with our output.

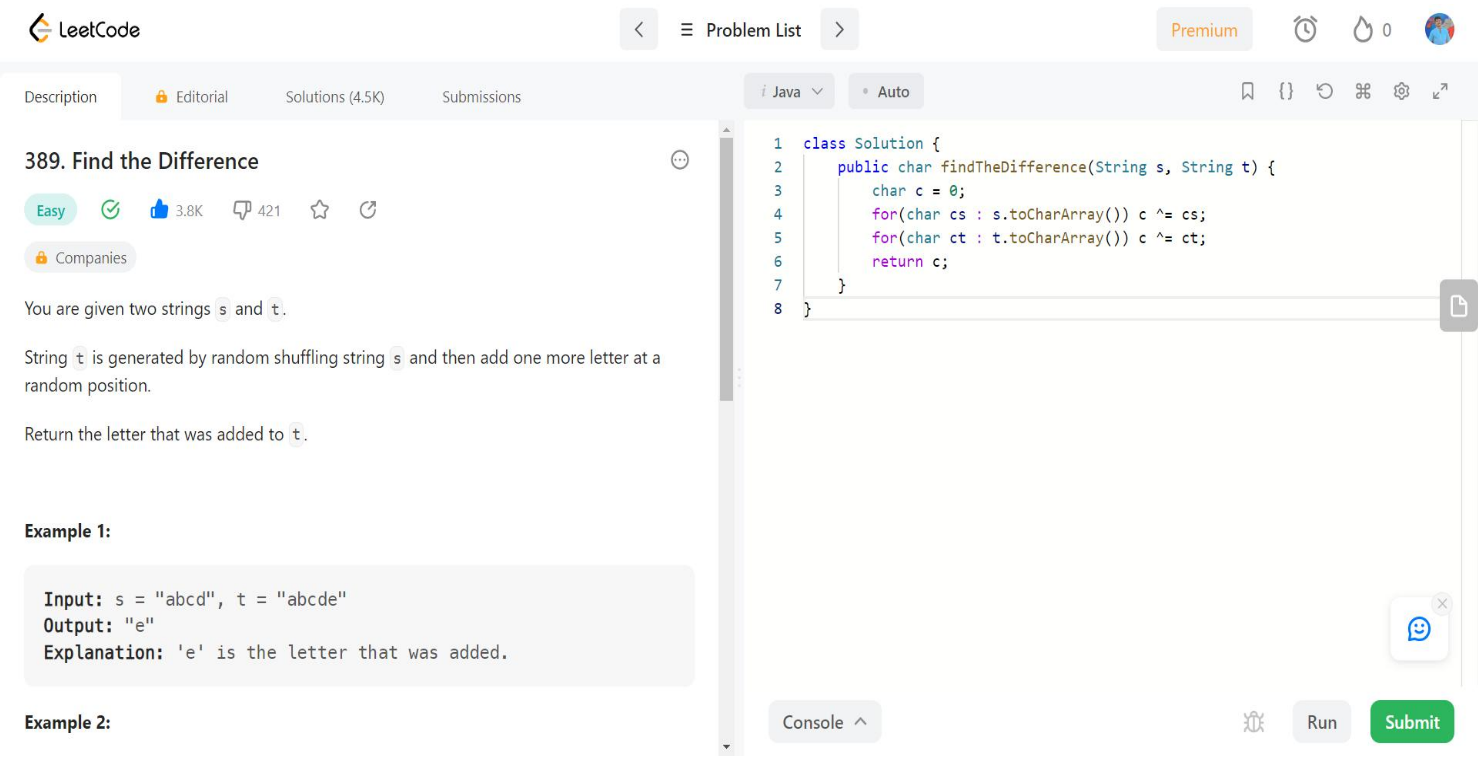
4. Program

I) . Find the Difference

You are given two strings s and t.
String t is generated by random shuffling string s and then add one more letter at a random position. Return the letter that was added to t.

CODE

```java
class Solution {
    public char findTheDifference(String s, String t) {
        char c = 0;
        for(char cs : s.toCharArray()) c ^= cs;
        for(char ct : t.toCharArray()) c ^= ct;
        return c;
    }
}
```



389. Find the Difference

You are given two strings s and t.

String t is generated by random shuffling string s and then add one more letter at a random position.

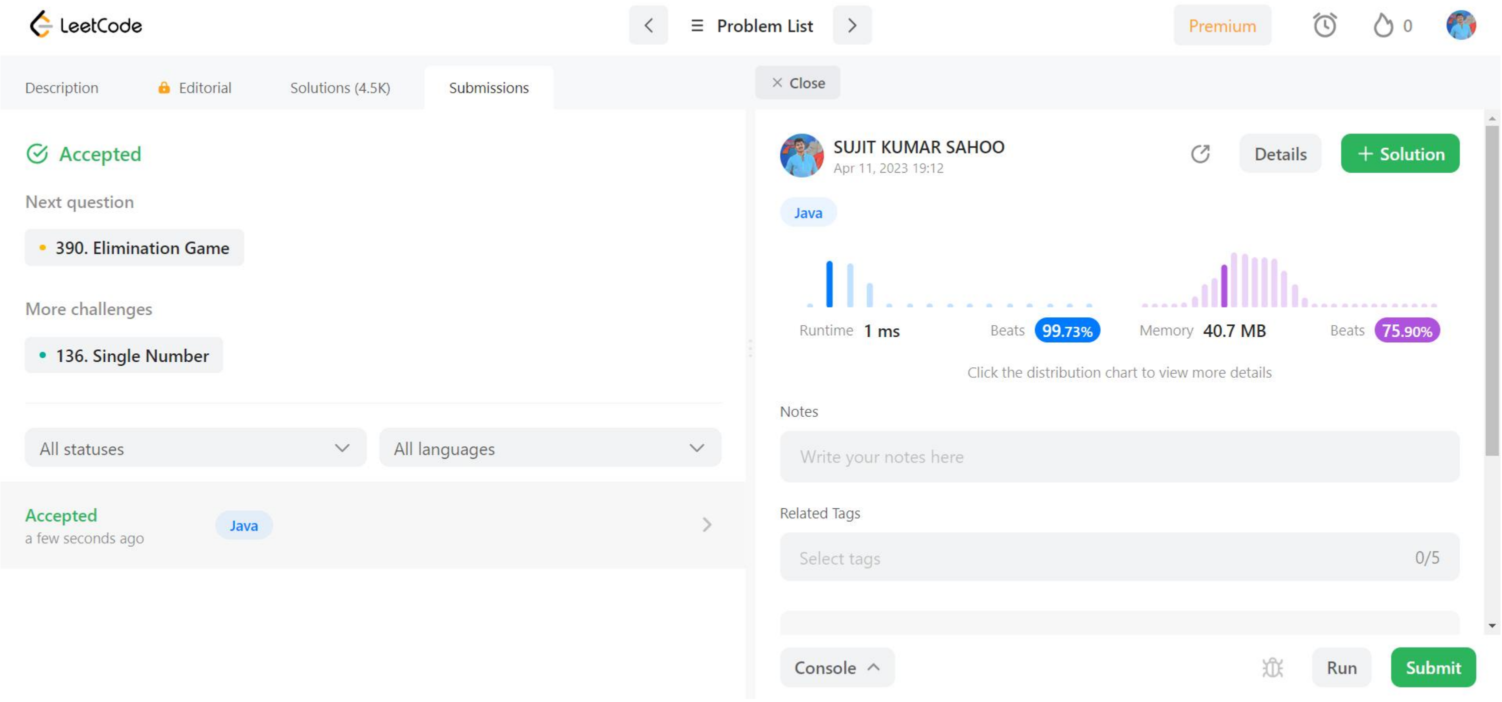
Return the letter that was added to t.

Example 1:

```
Input: s = "abcd", t = "abcde"
Output: "e"
Explanation: 'e' is the letter that was added.
```

Example 2:

```java
class Solution {
    public char findTheDifference(String s, String t) {
        char c = 0;
        for(char cs : s.toCharArray()) c ^= cs;
        for(char ct : t.toCharArray()) c ^= ct;
        return c;
    }
}
```



Accepted

Next question

- 390. Elimination Game

More challenges

- 136. Single Number

All statuses | All languages

Accepted — a few seconds ago — Java

SUJIT KUMAR SAHOO
Apr 11, 2023 19:12

Java

Runtime 1 ms — Beats 99.73%

Memory 40.7 MB — Beats 75.90%

Click the distribution chart to view more details

Notes

Write your notes here

Related Tags

Select tags 0/5

II) Gray Code

An n-bit gray code sequence is a sequence of 2n integers where:

Every integer is in the inclusive range [0, 2n - 1],

The first integer is 0,

An integer appears no more than once in the sequence,

The binary representation of every pair of adjacent integers differs by exactly one bit, and

The binary representation of the first and last integers differs by exactly one bit.

Given an integer n, return any valid n-bit gray code sequence.

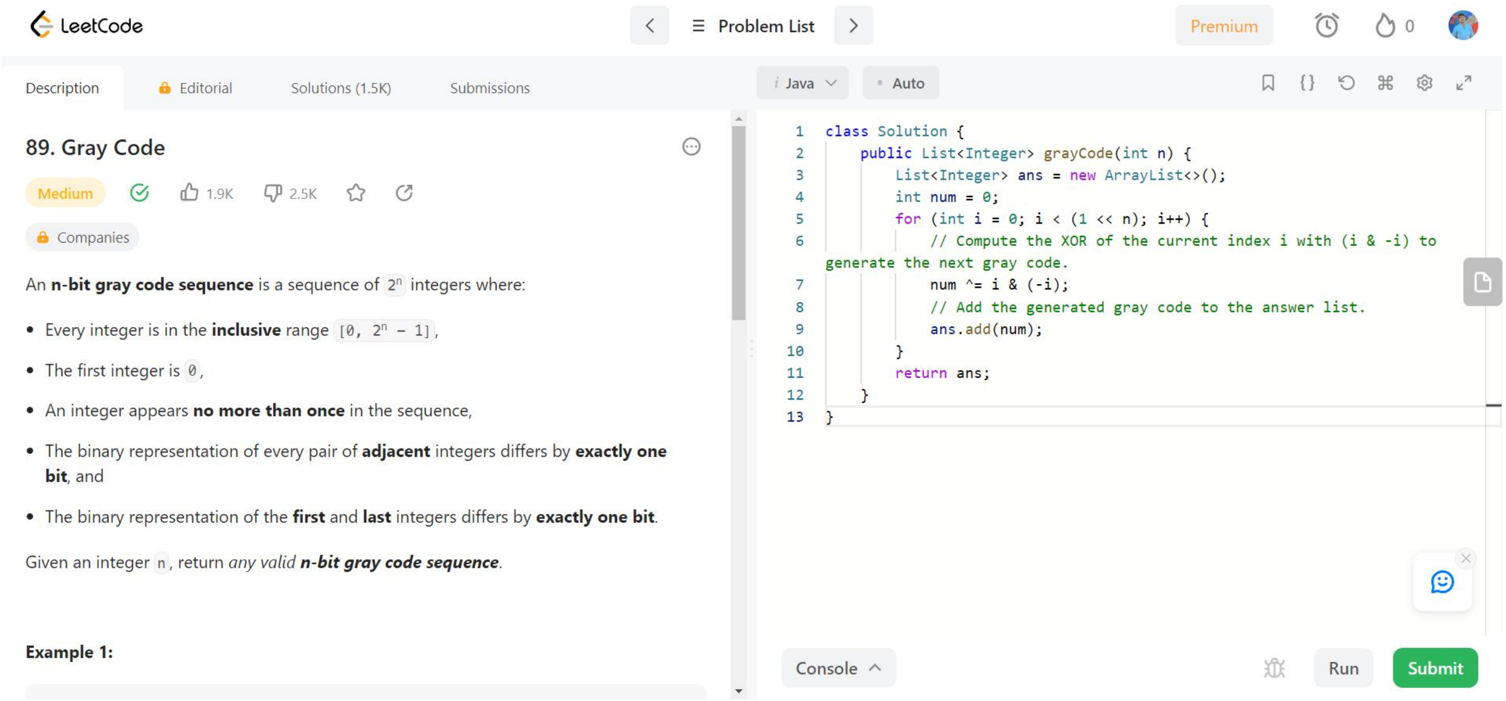
Algorithm:

1. We start with a list that contains only 0 as the first element.
Then, we iterate from 0 to (2^n - 1) and perform the following steps:
2. We compute the XOR of the current index i with (i & -i).
This operation flips the least significant bit that is set in i.
For example, if i = 3 (011), then (i & -i) = 1 (001).
So, num = num ^ (i & -i) will set the least significant bit of num to 1 if it is 0, and vice versa.
3. We add the computed num to the answer list.
4. Repeat steps 1 and 2 for all indices from 0 to (2^n - 1).

CODE

```
class Solution {
    public List<Integer> grayCode(int n) {
        List<Integer> ans = new ArrayList<>();
        int num = 0;
        for (int i = 0; i < (1 << n); i++) {
            // Compute the XOR of the current index i with (i & -i) to
generate the next gray code.
            num ^= i & (-i);
            // Add the generated gray code to the answer list.
            ans.add(num);
        }
        return ans;
    }
}
```

Output



89. Gray Code

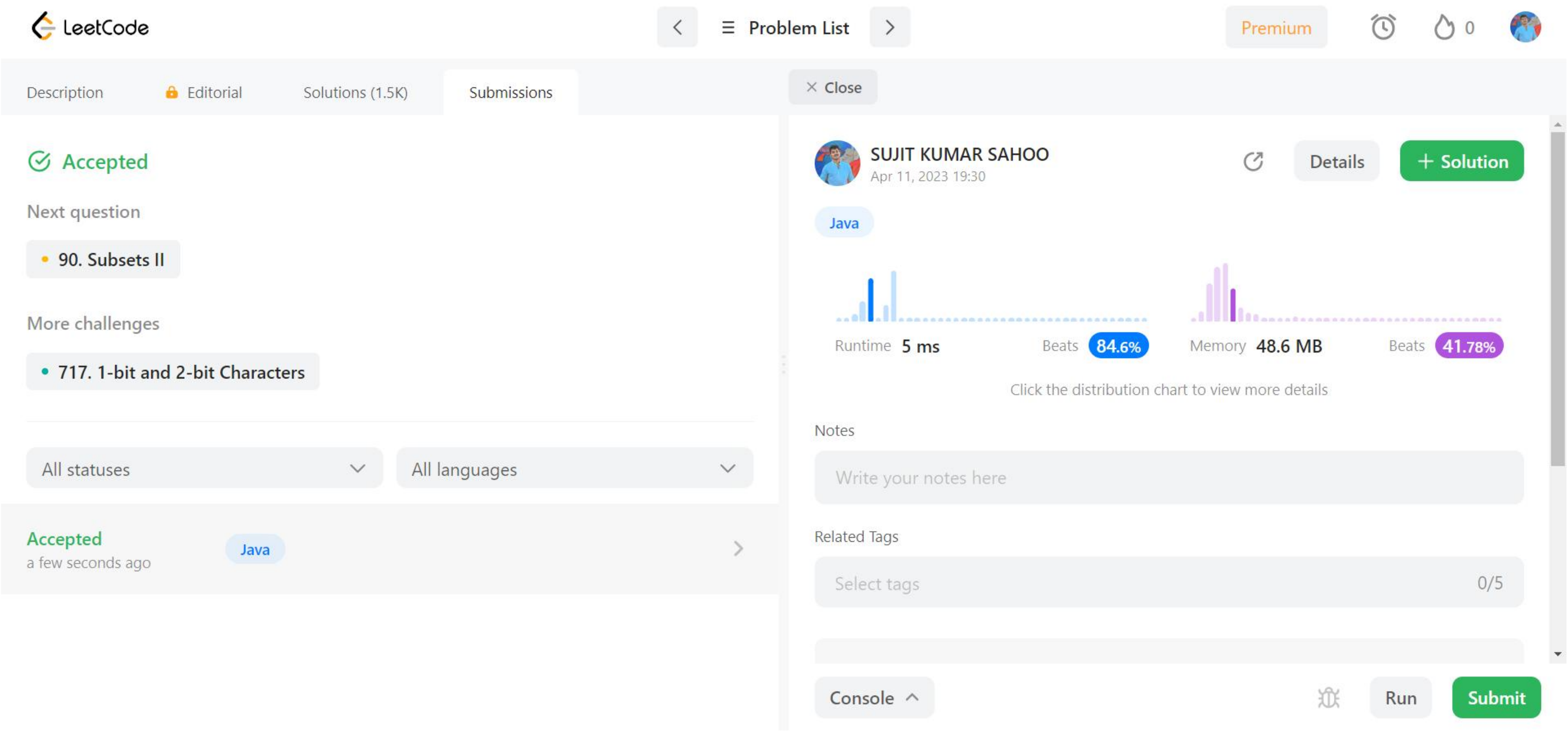
An **n-bit gray code sequence** is a sequence of 2^n integers where:

- Every integer is in the **inclusive** range `[0, 2^n - 1]`,
- The first integer is `0`,
- An integer appears **no more than once** in the sequence,
- The binary representation of every pair of **adjacent** integers differs by **exactly one bit**, and
- The binary representation of the **first** and **last** integers differs by **exactly one bit**.

Given an integer `n`, return *any valid* ***n*-bit gray code sequence**.

Example 1:

```java
class Solution {
    public List<Integer> grayCode(int n) {
        List<Integer> ans = new ArrayList<>();
        int num = 0;
        for (int i = 0; i < (1 << n); i++) {
            // Compute the XOR of the current index i with (i & -i) to generate the next gray code.
            num ^= i & (-i);
            // Add the generated gray code to the answer list.
            ans.add(num);
        }
        return ans;
    }
}
```



Accepted

Next question

90. Subsets II

More challenges

717. 1-bit and 2-bit Characters

All statuses | All languages

Accepted — a few seconds ago — Java

SUJIT KUMAR SAHOO
Apr 11, 2023 19:30

Java

Runtime 5 ms Beats 84.6%

Memory 48.6 MB Beats 41.78%

Notes

Write your notes here

Related Tags

Select tags 0/5