Find the Difference

Given two strings ***s*** and ***t*** which consist of only lowercase letters.

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

Find the letter that was added in ***t***.

Example:

```
Input:
s = "abcd"
t = "abcde"

Output:
e

Explanation:
'e' is the letter that was added.
```

Solution 1

```
public char findTheDifference(String s, String t) {
 char c = 0;
 for (int i = 0; i < s.length(); ++i) {
  c ^= s.charAt(i);
 }
 for (int i = 0; i < t.length(); ++i) {
  c ^= t.charAt(i);
 }
 return c;
}
```

maybe a more elegant version:

```
public char findTheDifference(String s, String t) {
 int n = t.length();
 char c = t.charAt(n - 1);
 for (int i = 0; i < n - 1; ++i) {
  c ^= s.charAt(i);
  c ^= t.charAt(i);
 }
 return c;
}
```

written by Harvery original link here

Solution 2

It is the same idea with 136. Single Number (https://leetcode.com/problems/single-number/)

```
class Solution {
public:
    char findTheDifference(string s, string t) {
        char r=0;
        for(char c:s) r ^=c;
        for(char c:t) r ^=c;
        return r;
    }
};
```

written by yanchao_hust original link here

Solution 3

Hi. There are several methods you can try to solve this. HashMap, Arrays, Bits, etc. Here, we're going to use a simple array of size 26 for alphabets. Then for each character in **s**, increment the count.

Then for each character in **t**, you should decrement the count. Now if at any point, the count goes below 0, then the character isn't present in **t**

```
        for (int i = 0; i < 26; i++) alpha[i] = 0;
        for (char c : s.toCharArray())
            alpha[ c - 'a' ]++;

        for (char c : t.toCharArray()) {
           //could do decrement first, then check but yeah
            if (--alpha[c - 'a'] < 0)
                return c;
        }

        return 0;
```

written by sharks12 original link here