Find the difference

Question:

https://leetcode.com/problems/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:

```
Input: s = "", t = "y"
Output: "y"
```

Approach 1:

I initially used a single hashmap to store the occurrence of each character of the string s and then checked it for each character of t.

It worked for unique extra character but failed at repetitions such as:

s="a"

t="aa"

Solution 1:

```
def findTheDifference(self,s,t):
        d=set()
        n=len(s)
        for i in range(n):
                if s[i] not in d:
                        d.add(s[i])
        for i in t:
                if i not in d:
                        return i
```

Approach 2:

To overcome this problem, I used two hashmaps.

One for string s and the other for string t.

And counted occurrences of each character.

And finally looped and checked if any key from hashmap **t** is not present in hashmap **s** or if the key had unequal counts.

This approach was accepted.

Solution 2:

```
def findTheDifference(self,s,t):
        d1={}
        d2={}
        n=len(t)
        for i in range(n):
                if i<n-1:
                        if s[i] not in d1:
                                d1[s[i]]=1
                        else:
                                d1[s[i]]+=1
                if t[i] not in d2:
                        d2[t[i]]=1
                else:
                        d2[t[i]]+=1
        for i in d2:
                if (i not in d1) or (d1[i]!=d2[i]):
                        return i
```

Time Complexity: O(n)

Space Complexity: O(n)

Approach 3:

I was scrolling through the discussions forum and found this bit manipulation approach using XOR.

Link to the detailed explanation:
https://leetcode.com/problems/find-the-difference/discuss/1751380/JavaC%2B%2BPython-very-very-EASY-to-go-solution

Solution 3:

```
def findTheDifference(self,s,t):
        n=len(s)
        c=0
        for i in s:
                c^=ord(i)

        for i in t:
                c^=ord(i)

        return chr(c)
```

Time Complexity: O(n)

Space Complexity: O(1)