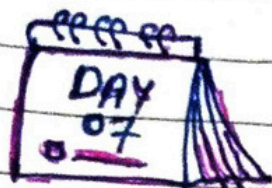


DAY 07/30 - #30DaysLogicBuilding Challenge.

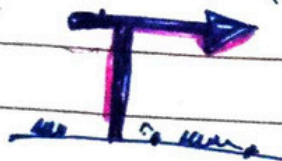
TODAY'S CHECKLIST ☑



→ Solving "STRING" Anagram problem



Question



"Valid Anagram"

"Anagram - Rearranging letters of a word to create a new one."

Given Word → $S_1 = \text{race}$, $S_2 = \text{care}$

↓
return true → if it is correct

Pseudocode -

FUNCTION isAnagram (S_1, S_2):

IF length of S_1 is not equal to S_2 :

RETURN false

CREATE array f_1 of size 26 and initialize to 0 -

FOR each character ch in S_1 :

INCREMENT $f_1[\text{ch} - 'a']$ by 1

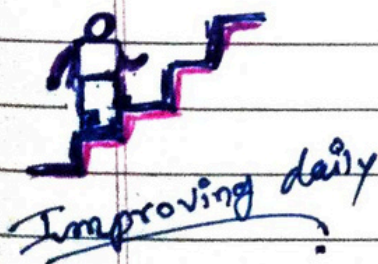
create f_2 array and do the same as above

FOR idx from 0 to 25:

IF $f_1[\text{idx}]$ is not equal to $f_2[\text{idx}]$:

RETURN false.

RETURN true.



242. Valid Anagram

Easy Topics Companies

Given two strings `s` and `t`, return `true` if `t` is an anagram of `s`, and `false` otherwise.

An **Anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

Example 1:
Input: `s = "anagram", t = "nagaram"`
Output: `true`

Example 2:
Input: `s = "rat", t = "car"`
Output: `false`

12.1K 194

```
Java Auto
int[] f1 = new int[26];
for(int i=0; i<s1.length(); i++){
    char ch = s1.charAt(i);
    f1[ch - 'a']++;
}

int[] f2 = new int[26];
for(int i=0; i<s2.length(); i++){
    char ch = s2.charAt(i);
    f2[ch - 'a']++;
}

for(int i=0; i<26; i++){
    if(f1[i] != f2[i]){
        return false;
    }
}
```

Accepted Runtime: 0 ms

Case 1 Case 2





Output Window

[Compilation Results](#) Custom Input

[Suggest Feedback](#)

Compilation Completed

For Input:  

race care

Your Output:

YES

Expected Output:

YES

Java (1.8)

Average Time: 20m

[Start Timer](#)

```
1 // } Driver Code Ends
36
37
38 class Solution
39 {
40     //Function is to check whether two strings are anagram of each other or not.
41     public static boolean isAnagram(String a,String b)
42     {
43
44         // Your code here
45         int f1[] = new int[26];
46         for(int i=0; i<a.length(); i++){
47             char ch = a.charAt(i);
48             f1[ch - 'a']++;
49         }
50
51         int f2[] = new int[26];
52         for(int i=0; i<b.length(); i++){
53             char ch = b.charAt(i);
54             f2[ch - 'a']++;
55         }
56
57         for(int i=0; i<26; i++){
58             if(f1[i] != f2[i]){
59                 return false;
60             }
61         }
62         return true;
63     }
64 }
```



[Custom Input](#)

[Compile & Run](#)

[Submit](#)

