242. Valid Anagram

Created	@January 29, 2022 2:31 PM
Difficulty	Easy
LC Url	https://leetcode.com/problems/valid-anagram/
Importance	
Tag	Array&Sorting
Video	https://www.youtube.com/watch?v=wScXoa8pN6o&list=PL2rWx9cCzU85RX9NeRMVUV_kgI4YGKURD&index=51

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
```

Constraints:

- `1 <= s.length, t.length <= 5 * 10 4`
- `s` and `t` consist of lowercase English letters.

Follow up: What if the inputs contain Unicode characters? How would you adapt your solution to such a case?

Solution

```
class Solution:
    def isAnagram(self, s: str, t: str) -> bool:
        '''
        时间空间复杂度都是O(n)
        '''
        if len(s) != len(t):
            return False

        lookup = {}

        for i in s:
            if i not in lookup:
                lookup[i] = 1
            else:
                lookup[i] += 1

        for j in t:
            if j not in lookup:
                return False
            else:
                lookup[j] -= 1

        for k in lookup:
            if lookup[k] != 0:
                return False

        return True
```

```
class Solution:
    def isAnagram(self, s: str, t: str) -> bool:
        if len(s) != len(t):
            return False

        countS, countT = {}, {}

        for i in range(len(s)):
            countS[s[i]] = 1 + countS.get(s[i], 0)
            countT[t[i]] = 1 + countT.get(t[i], 0)
        return countS == countT
```