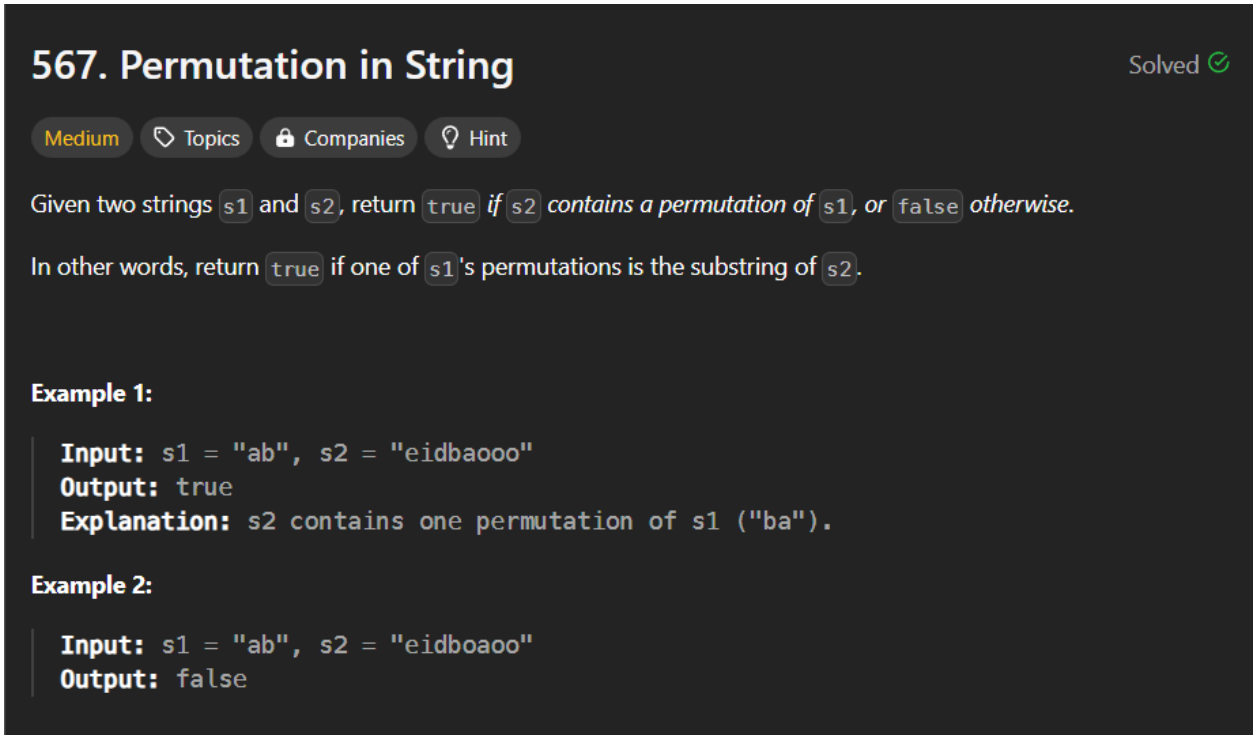
3: Leetcode 567: Permutation in String

⇒ https://leetcode.com/problems/permutation-in-string/description/



567. Permutation in String

Solved

Medium | Topics | Companies | Hint

Given two strings `s1` and `s2`, return `true` *if* `s2` *contains a permutation of* `s1`*, or* `false` *otherwise.*

In other words, return `true` if one of `s1`'s permutations is the substring of `s2`.

Example 1:

```
Input: s1 = "ab", s2 = "eidbaooo"
Output: true
Explanation: s2 contains one permutation of s1 ("ba").
```

Example 2:

```
Input: s1 = "ab", s2 = "eidboaoo"
Output: false
```

A: PYTHON doubts:

→

- getting ascii value , A = 65, a = 97, 1 = 49 ⇒ **ord('a') = 97**

B: Code -

→

```python
class Solution:
    def checkInclusion(self, s1: str, s2: str) -> bool:
        n1 = len(s1)
        n2 = len(s2)

        if n1 > n2:
            return False

        s1Count = [0] * 26
        s2Count = [0] * 26

        for i in range(n1):
            s1Count[ord(s1[i]) - ord('a')] += 1
            s2Count[ord(s2[i]) - ord('a')] += 1
```

```
        if s1Count == s2Count:
            return True

        for i in range(n1,n2):
            s2Count[ord(s2[i]) - ord('a')] += 1
            s2Count[ord(s2[i - n1]) - ord('a')] -= 1
            if s1Count == s2Count:
                return True

        return False
```

C: Written notes

Q) 567 : Permutations in String :
→ YT : Greg Hogg (567)

A) Algo / Approach :
⇒ Brute force → calc all permutatⁿ of s1 → n! (len(s1)) & compare
⇒ TLE

*) Count + Sliding window: → counts [s2] = [0 - - 25]
s1 = 'ab'   s2 = 'eidbaoo' → only this (len(s1) window). → size 26
counts [s1] = [1 1 0 0 . . 0]
0 1 2 - - 25
a b

Initial compare, false then go ahead → 0,1 checked
n1 = len(s1) = 2,
start from → n = 2 till n2 = len(s2)
start 2,3,4

if false → inc count of s2[n1] → s2[2]
dec count of s2[i-n1] → s2[2-2] = s[0]
again compare.

B) Code:
→ 1) check edge case : len(s1) > len(s2) → F
2) Initialize count array → 26 size (Alphabet that also only lower).
3) Initial count array till n1
4) use ord → gives ascii code value.
4) Compare initial, if F then continue
5) Else → start from n1 to n2.
inc count = s2 [n1]
dec count = s2 [i - n1].
6) compare s1, s2 → Equal then True
Else False.