#392

Is Subsequence

Akhtamov Azimjon **2024299010**

2025-01-23

Problem Definition (1)

- **Source**: Leetcode

- Title: **Is Subsequence**

- Difficulty: **easy**

- Type: **Strings**

Problem Definition (1)

392. Is Subsequence

Easy · Topics · Companies

Given two strings `s` and `t`, return `true` if `s` is a **subsequence** of `t`, or `false` otherwise.

A **subsequence** of a string is a new string that is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (i.e., `"ace"` is a subsequence of `"abcde"` while `"aec"` is not).

Example 1:

```
Input: s = "abc", t = "ahbgdc"
Output: true
```

Example 2:

```
Input: s = "axc", t = "ahbgdc"
Output: false
```

Constraints:

- `0 <= s.length <= 100`
- `0 <= t.length <= 10^4`
- `s` and `t` consist only of lowercase English letters.
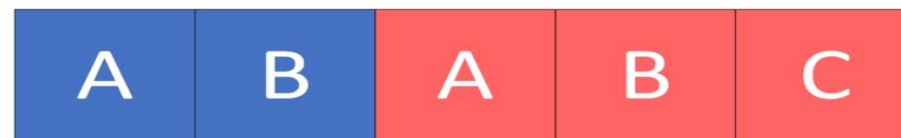
Solution (1):

s = "THE"   t = "TUREHEM"

j   i   *#times*

Solution (2):

```python
class Solution:
    def isSubsequence(self, s: str, t: str) -> bool:
        S = len(s)
        T = len(t)

        if s == '': return True
        if S > T: return False

        j = 0
        for i in range(T):
            if t[i] == s[j]:
                if j == S-1: return True
                j += 1
        return False
```
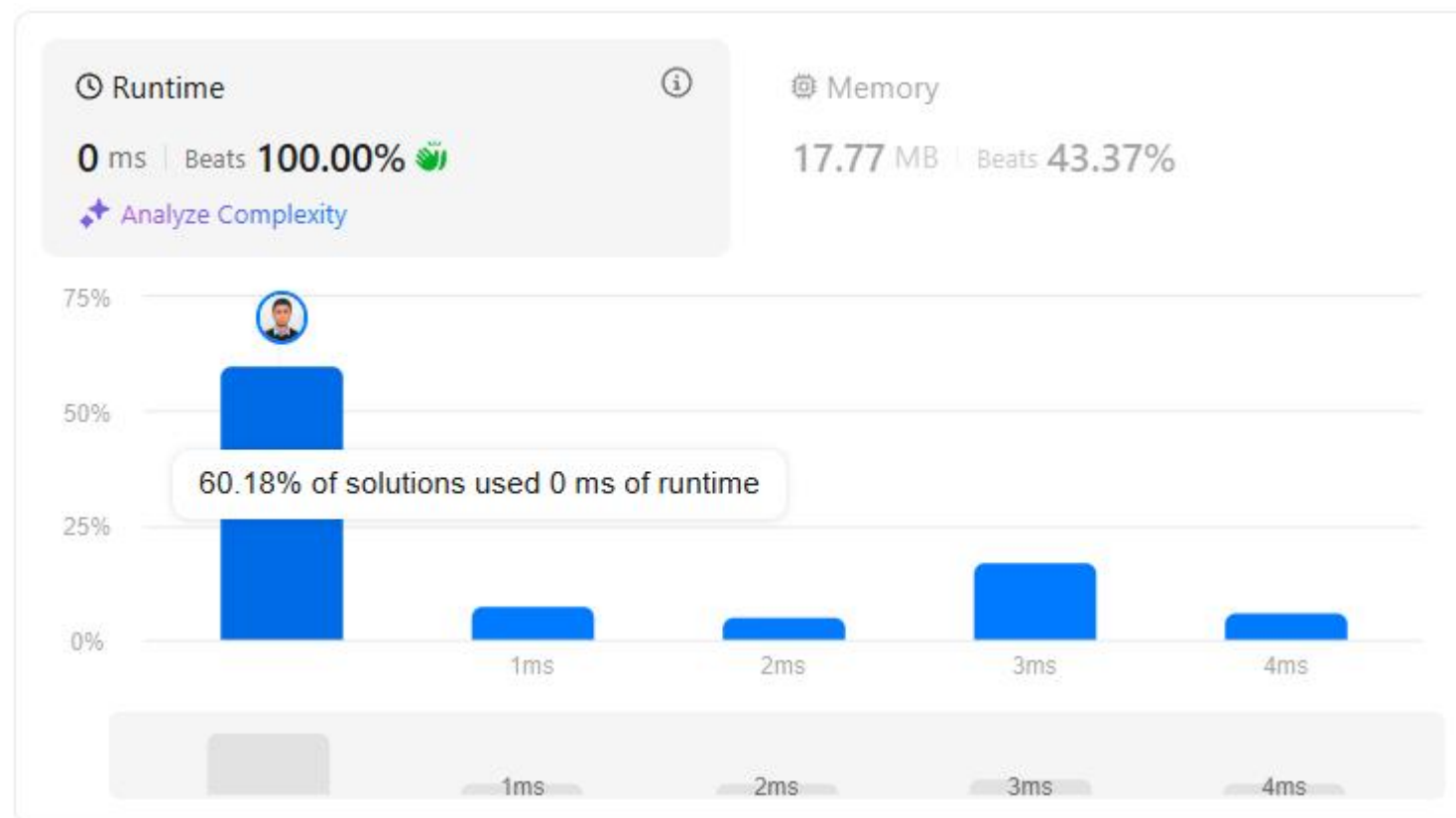
*# **len()** function to get time to check operator*

s is nothing , it means s is always subsequence of t
if lenght of t < s, there is no option

#checking based on t[i] = s[j]

Solution (3)

What I have learned

❖**Strings:**

✓Checking Strings based on given opeartors

✓Time: O(T) # T times for the length of s(string)
✓Space: O(1) #no space

Questions and Answers

Greetings