392. Is Subsequence

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 <= 104`
- `s` and `t` consist only of lowercase English letters.

Follow up: Suppose there are lots of incoming `s`, say `s1, s2, ..., sk` where `k >= 109`, and you want to check one by one to see if `t` has its subsequence. In this scenario, how would you change your code?

```python
class Solution:
    def isSubsequence(self, s: str, t: str) -> bool:
        if len(s)==0:
            return True
        if len(t)==0:
            return False
        i = 0
        j = 0
        while i<len(t):
            if s[j]==t[i]:
                j = j+1
            if j==len(s):
                return True
```

```
            i = i+1
        return False
```