392 Is Subsequence (link)

Description

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.

Follow up: Suppose there are lots of incoming `s`, say $s_1, s_2, ..., s_k$ where $k >= 10^9$, and you want to check one by one to see if `t` has its subsequence. In this scenario, how would you change your code?

(scroll down for solution)

Solution

Language: cpp

Status: Accepted

```cpp
#include <string>

using namespace std;

class Solution {
public:
    bool isSubsequence(string s, string t) {
        int i = 0; // Индекс для строки s
        int j = 0; // Индекс для строки t

        // Пока не достигли конца строк s или t
        while (i < s.length() && j < t.length()) {
            // Если символы совпадают, двигаем указатель i
            if (s[i] == t[j]) {
                i++;
            }
            // Всегда двигаем указатель j
            j++;
        }

        // Если i достигло конца строки s, значит s является подпоследовательностью t
        return i == s.length();
    }
};
```