392. Is Subsequence

Given two strings `s` and `t`, check if `s` is a **subsequence** of `t`.

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.```Python
  def isSubsequence(self, s: str, t: str) -> bool:
  dp = [[0 for j in range(len(t)+1)] for i in range(len(s)+1)]

```
        for i in range(1,len(s)+1):
            for j in range(1,len(t)+1):
                if s[i-1]==t[j-1]:
                    dp[i][j] = 1+dp[i-1][j-1]
                else:
                    dp[i][j] = max(dp[i-1][j],dp[i][j-1])
        return dp[len(s)][len(t)]==len(s)

    #Second Approach
        i = 0
        j=0
        while i<len(s) and j<len(t):
            if s[i]==t[j]:
                i = i+1
            j = j+1
        return i==len(s)
```