# 判断子序列

给定字符串 **s** 和 **t** ，判断 **s** 是否为 **t** 的子序列。

字符串的一个子序列是原始字符串删除一些（也可以不删除）字符而不改变剩余字符相对位置形成的新字符串。（例如，"ace"是"abcde"的一个子序列，而"aec"不是）。

**进阶：**

如果有大量输入的 S，称作 S1, S2, ... , Sk 其中 k >= 10 亿，你需要依次检查它们是否为 T 的子序列。在这种情况下，你会怎样改变代码？

**致谢：**

特别感谢 @pbrother 添加此问题并且创建所有测试用例。

**示例 1：**

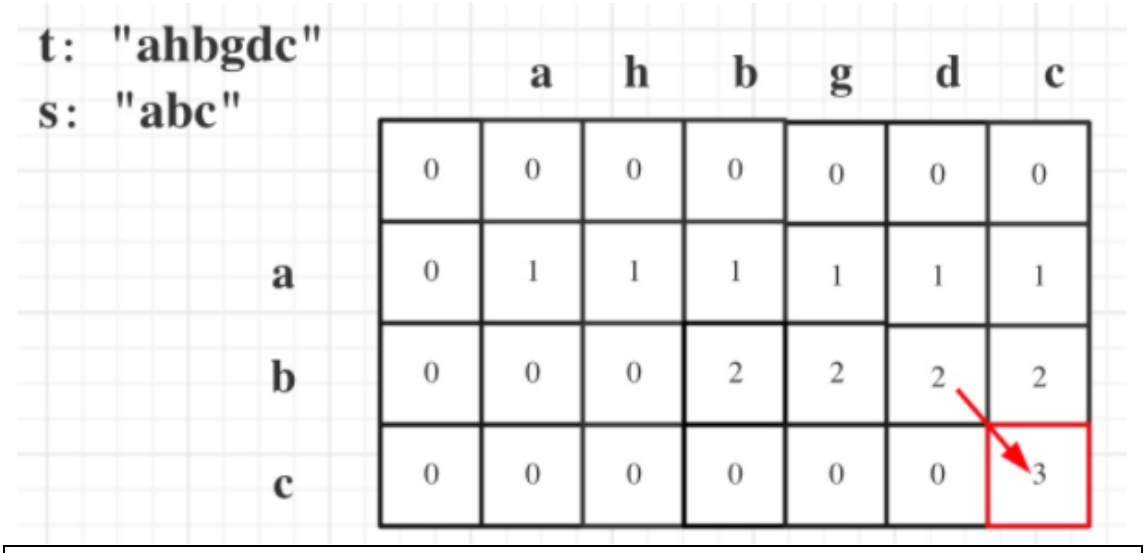
**输入：** `s = "abc", t = "ahbgdc"`

**输出：** `true`

**示例 2：**

**输入：** `s = "axc", t = "ahbgdc"`

**输出：** `false`




```
class Solution {
public:
    bool isSubsequence(string s, string t) {
        vector<vector<int>> dp(s.size()+1,vector<int>(t.size()+1));
```

```
        int i,j;
        for (i = 1; i <= s.size(); i++)
        {
            for (j = 1; j <= t.size(); j++)
            {
                if (s[i-1] == t[j-1])
                {
                    dp[i][j] = dp[i - 1][j - 1] + 1;
                }
                else
                {
                    dp[i][j] = max(dp[i][j-1],dp[i-1][j]);
                }

            }
        }
        if(dp[s.size()][t.size()]==s.size())//判断最长公共子串与 s 长度相同则返回 true
            return true;
        else
            return false;
    }
};
```