# 题目

给定字符串s和t，判断s是否为t的子序列。

你可以认为s和t中仅包含英文小写字母。字符串t可能会很长（长度 ~= 500,000），而s是个短字符串（长度<=100）。

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）。

**示例 1:**

s = "abc", t = "ahbgdc"

返回 true.

**示例 2:**

s = "axc", t = "ahbgdc"

返回 false.

**后续挑战 :**

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？

# 分析

## 方法一：双指针

**思路：**

可以知道的是，当扫描s中的第k个字符时，假如它在t字符串中的第i位和第j位都出现过(i < j)，那么我们从左到右扫描到第i位时，就认为已经找到了s中第k个字符。因为i后面有更多的备选字符可以用来找s中的剩余字符。也就是说，我们在t中找字符时，是严格不回溯的。这个问题可以使用双指针解决。

初始化指针i，j为0，分别指向s和j的第0个字符，在t中找到s[i]字符后，i++试图找下一个字符。

若最后i到达s末尾，则说明找到了该字符串。

**代码：**

class Solution {

public:

bool isSubsequence(string s, string t) {

int i = 0, j = 0;

int m = s.size(), n = t.size();

while(i < m && j < n){

if(s[i] == t[j]){

i++;

j++;

}

else j++;

}

return i == m;

}

};

方法二：