

Given a string **s** and a string **t**, check if **s** is subsequence of **t**.

You may assume that there is only lower case English letters in both **s** and **t**. **t** is potentially a very long (length \sim 500,000) string, and **s** is a short string (≤ 100).

A subsequence of a string is a new string which is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (ie, "ace" is a subsequence of "abcde" while "aec" is not).

Example 1:

s = "abc", **t** = "ahbgdc"

Return true.

Example 2:

s = "axc", **t** = "ahbgdc"

Return false.

```
public class L392 {
    /*
     * 判断s是不是t的子串，如果是返回true，不是返回false
     */
    public boolean isSubsequence(String s, String t) {

        if(t== null || t.length() < s.length())
            return false;

        if(t.length() == 0 && s.length() == 0)
            return true;

        int i = 0;
        int j = 0;
        while(i < s.length() && j < t.length()) {
            if(s.charAt(i) == t.charAt(j)) {
                i ++;
                j ++;
            }else {
                j ++;
            }
        }

        if(j < t.length() || (i == s.length() && j == t.length()))
            return true;
        else
            return false;
    }
}
```