Is Subsequence

Question body:

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 ~= 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.

Key:

The idea of this question is quite trivial. One can easily come up with an O(n) algorithm to satisfy the purpose. However, to get the solution completely correct, one really need to pay attention to the corner case. In my first attempt, I missed the corner case “s = empty string” and the case there is multiple ending character for S in T.

Solution:

public class Solution {

public boolean isSubsequence(String s, String t) {

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

int index=0;

int expected\_index= s.length();

char[] sSet = s.toCharArray();

char[] tSet = t.toCharArray();

for(int pos=0; pos< t.length(); pos++){

if(tSet[pos]==sSet[index]){

index++;

}

if(index == expected\_index){

return true;

}

}

if(index == expected\_index) return true;

return false;

}

}