Find the smallest window in a string containing all characters of another string

Given two strings string1 and string2, find the smallest substring in string1 containing all characters of string2 efficiently.

For Example:

Input string1: “this is a test string”
Input string2: “tist”
Output string: “t stri”

We strongly recommend that you click here and practice it, before moving on to the solution.

Method 1 (Brute force solution)
a) Generate all substrings of string1 (“this is a test string”)
b) For each substring, check whether the substring contains all characters of string2 (“tist”)
c) Finally print the smallest substring containing all characters of string2.

Method 2 (Efficient Solution)
1) Build a count array count[] for string 2. The count array stores counts of characters.
count[‘i’] = 1
count[‘t’] = 2
count[‘s’] = 1

2) Scan the string1 from left to right until we find all the characters of string2. To check if all the characters are there, use count[] built in step 1. So we have substring “this is a t” containing all characters of string2. Note that the first and last characters of the substring must be present in string2. Store the length of this substring as min_len.

3) Now move forward in string1 and keep adding characters to the substring “this is a t”. Whenever a character is added, check if the added character matches the left most character of substring. If matches, then add the new character to the right side of substring and remove the leftmost character and all other extra characters after left most character. After removing the extra characters, get the length of this substring and compare with min_len and update min_len accordingly.

Basically we add ‘e’ to the substring “this is a t”, then add ‘s’ and then’t’. ‘t’ matches the left most character, so remove ‘t’ and ‘h’ from the left side of the substring. So our current substring becomes “is a test”. Compare length of it with min_len and update min_len.
Again add characters to current substring “is a test”. So our string becomes “is a test str”. When we add ‘i’, we remove leftmost extra characters, so current substring becomes “t stri”. Again, compare length of it with min_len and update min_len. Finally add ‘n’ and ‘g’. Adding these characters doesn’t decrease min_len, so the smallest window remains “t stri”.

4) Return min_len.

Please write comments if you find the above algorithms incorrect, or find other ways to solve the same problem.