Our Solution(s)

Run Code

Solution 1

```javascript
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

// O(b + s) time | O(b + s) space - where b is the length of the big
// input string and s is the length of the small input string
function smallestSubstringContaining(bigString, smallString) {
  const targetCharCounts = getCharCounts(smallString);
  const substringBounds = getSubstringBounds(bigString, targetCharCou
  return getStringFromBounds(bigString, substringBounds);
}

function getCharCounts(string) {
  const charCounts = {};
  for (const char of string) {
    increaseCharCount(char, charCounts);
  }
  return charCounts;
}

function getSubstringBounds(string, targetCharCounts) {
  let substringBounds = [0, Infinity];
  const substringCharCounts = {};
  const numUniqueChars = Object.keys(targetCharCounts).length;
  let numUniqueCharsDone = 0;
  let leftIdx = 0;
  let rightIdx = 0;
  // Move the rightIdx to the right in the string until you've counted
  // all of the target characters enough times.
  while (rightIdx < string.length) {
    const rightChar = string[rightIdx];
    if (!(rightChar in targetCharCounts)) {
      rightIdx++;
      continue;
    }
```

Your Solutions

Run Code

Solution 1   Solution 2   Solution 3

```javascript
function smallestSubstringContaining(bigString, smallString) {
  // Write your code here.
}

// Do not edit the line below.
exports.smallestSubstringContaining = smallestSubstringContaining;
```

Our Tests

Custom Output

Submit Code

Run or submit code when you're ready.

Run or submit code when you're ready.