Our Solution(s) Run Code

Solution 1 Solution 2 Solution 3

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(ns + bs) time | O(ns) space
def multiStringSearch(bigString, smallStrings):
    trie = Trie()
    for string in smallStrings:
        trie.insert(string)
    containedStrings = {}
    for i in range(len(bigString)):
        findSmallStringsIn(bigString, i, trie, containedStrings)
    return [string in containedStrings for string in smallStrings]


def findSmallStringsIn(string, startIdx, trie, containedStrings):
    currentNode = trie.root
    for i in range(startIdx, len(string)):
        currentChar = string[i]
        if currentChar not in currentNode:
            break
        currentNode = currentNode[currentChar]
        if trie.endSymbol in currentNode:
            containedStrings[currentNode[trie.endSymbol]] = True


class Trie:
    def __init__(self):
        self.root = {}
        self.endSymbol = "*"

    def insert(self, string):
        current = self.root
        for i in range(len(string)):
            if string[i] not in current:
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```python
def multiStringSearch(bigString, smallStrings):
    # Write your code here.
    pass

```

Our Tests

Custom Output Submit Code

```
            program.multiStringSearch('this is a big string', ['this'],
            [True, False, True, True, False, True, False],
        )

    def test_case_2(self):
        self.assertEqual(
            program.multiStringSearch(
                'Mary goes to the shopping center every week.',
                ['he', 'Mary', 'centers', 'shop', 'shopping', 'string'
            ],
            [True, True, False, True, True, False, False],
        )

    def test_case_3(self):
        self.assertEqual(
```

Run or submit code when you're ready.