

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(bns) time | O(n) space
5     func multiStringSearch(_ bigString: String, _ smallStrings: [String]) -> Bool {
6         return smallStrings.map { isInBigString($0, bigString) }
7     }
8
9     func isInBigString(_ smallString: String, _ bigString: String) -> Bool {
10         for i in 0 ..< bigString.count {
11             if i + smallString.count > bigString.count {
12                 break
13             }
14
15             if isInBigStringHelper(i, smallString, bigString) {
16                 return true
17             }
18         }
19
20         return false
21     }
22
23     func isInBigStringHelper(_ startIndex: Int, _ smallString: String,
24                             _ bigString: String) -> Bool {
25         var leftSmallIndex = 0
26         var rightSmallIndex = smallString.count - 1
27
28         var leftBigIndex = startIndex
29         var rightBigIndex = startIndex + smallString.count - 1
30
31         while leftBigIndex < rightBigIndex {
32             let leftSmallStringIndex = smallString.index(smallString.startIndex, offsetBy: leftSmallIndex)
33             let rightSmallStringIndex = smallString.index(smallString.startIndex, offsetBy: rightSmallIndex)
```

```
1 class Program {
2     func multiStringSearch(_ bigString: String, _ smallStrings: [String]) -> Bool {
3         // Write your code here.
4         return []
5     }
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 class Program {
2     func multiStringSearch(_ bigString: String, _ smallStrings: [String]) -> Bool {
3         // Write your code here.
4         return []
5     }
6 }
7
```

```
1 class Program {
2     func multiStringSearch(_ bigString: String, _ smallStrings: [String]) -> Bool {
3         // Write your code here.
4         return []
5     }
6 }
7
```

```
10         return [None, None, None, None, None, None, None, None, None, None]
11     }
12     return [None, None, None, None, None, None, None, None, None, None]
13 }
14
15 // Example 2: [None, None, None, None, None, None, None, None, None, None]
16 // Expected: [None, None, None, None, None, None, None, None, None, None]
17 // Actual: [None, None, None, None, None, None, None, None, None, None]
18
19 // Example 3: [None, None, None, None, None, None, None, None, None, None]
20 // Expected: [None, None, None, None, None, None, None, None, None, None]
21 // Actual: [None, None, None, None, None, None, None, None, None, None]
22
23 // Example 4: [None, None, None, None, None, None, None, None, None, None]
24 // Expected: [None, None, None, None, None, None, None, None, None, None]
25 // Actual: [None, None, None, None, None, None, None, None, None, None]
26
27 // Example 5: [None, None, None, None, None, None, None, None, None, None]
28 // Expected: [None, None, None, None, None, None, None, None, None, None]
29 // Actual: [None, None, None, None, None, None, None, None, None, None]
```

Run or submit code when you're ready.