

Solution 1Solution 2Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(b ^ 2 + ns) time | O(b ^ 2 + n) space
5     func multiStringSearch(_ bigString: String, _ smallStrings: [String]) -> [Bool] {
6         let modifiedSuffixTrie = ModifiedSuffixTrie(string: bigString)
7
8         return smallStrings.map { modifiedSuffixTrie.contains($0) }
9     }
10
11     class TrieNode {
12         var children: [String: Any] = [:]
13     }
14
15     class ModifiedSuffixTrie {
16         var root = TrieNode()
17
18         init(string: String) {
19             populateModifiedSuffixTrieFrom(string)
20         }
21
22         func populateModifiedSuffixTrieFrom(_ string: String) {
23             for i in 0 ..< string.count {
24                 insertSubstringStartingAt(i, string, root)
25             }
26         }
27
28         func insertSubstringStartingAt(_ index: Int, _ string: String, _ root: TrieNode) {
29             var node = root
30
31             for j in index ..< string.count {
32                 let jStringIndex = string.index(string.startIndex, offsetBy: j)
33                 let jthCharacter = String(string[jStringIndex])
34
35                 if !node.children.keys.contains(jthCharacter) {
36                     node.children[jthCharacter] = TrieNode()
37                 }
38
39                 let nextNode = node.children[jthCharacter] as! TrieNode
40                 node = nextNode
41             }
42         }
43
44         func contains(_ string: String) -> Bool {
45             var node = root
46
47             for character in string {
48                 let stringifiedCharacter = String(character)
49
50                 if !node.children.keys.contains(stringifiedCharacter) {
51                     return false
52                 }
53
54                 let nextNode = node.children[stringifiedCharacter] as! TrieNode
55                 node = nextNode
56             }
57
58             return true
59         }
60     }
61 }
62
```

