Prompt   Scratchpad   Our Solution(s)   Video Explanation   Run Code

Solution 1   Solution 2   Solution 3

```java
1  // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  import java.util.*;
4
5  class Program {
6    // O(b^2 + ns) time | O(b^2 + n) space
7    public static List<Boolean> multiStringSearch(String bigString, String[] smallStrings) {
8      ModifiedSuffixTrie modifiedSuffixTrie = new ModifiedSuffixTrie(bigString);
9      List<Boolean> solution = new ArrayList<Boolean>();
10     for (String smallString : smallStrings) {
11       solution.add(modifiedSuffixTrie.contains(smallString));
12     }
13     return solution;
14   }
15
16   static class TrieNode {
17     Map<Character, TrieNode> children = new HashMap<Character, TrieNode>();
18   }
19
20   static class ModifiedSuffixTrie {
21     TrieNode root = new TrieNode();
22
23     public ModifiedSuffixTrie(String str) {
24       populateModifiedSuffixTrieFrom(str);
25     }
26
27     public void populateModifiedSuffixTrieFrom(String str) {
28       for (int i = 0; i < str.length(); i++) {
29         insertSubstringStartingAt(i, str);
30       }
31     }
32
33     public void insertSubstringStartingAt(int i, String str) {
34       TrieNode node = root;
35       for (int j = i; j < str.length(); j++) {
36         char letter = str.charAt(j);
37         if (!node.children.containsKey(letter)) {
38           TrieNode newNode = new TrieNode();
39           node.children.put(letter, newNode);
40         }
41         node = node.children.get(letter);
42       }
43     }
44
45     public boolean contains(String str) {
46       TrieNode node = root;
47       for (int i = 0; i < str.length(); i++) {
48         char letter = str.charAt(i);
49         if (!node.children.containsKey(letter)) {
50           return false;
51         }
52         node = node.children.get(letter);
53       }
54       return true;
55     }
56   }
57 }
58
```