Sublime

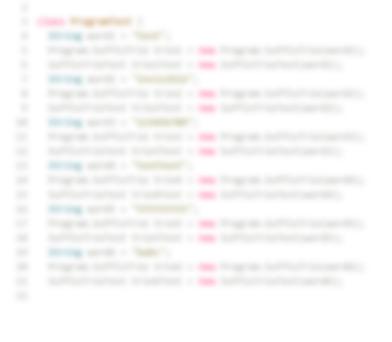
Our Solution(s)

Run Code

```
Your Solutions
```

```
Solution 1
 1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
   import java.util.*;
5 class Program {
6
     static class TrieNode {
       Map<Character, TrieNode> children = new HashMap<Character,</pre>
9
10
      static class SuffixTrie {
11
       TrieNode root = new TrieNode();
12
        char endSymbol = '*';
13
        public SuffixTrie(String str) {
14
15
         populateSuffixTrieFrom(str);
16
17
        // O(n^2) time | O(n^2) space
18
19
        public void populateSuffixTrieFrom(String str) {
20
          for (int i = 0; i < str.length(); i++) {</pre>
21
            insertSubstringStartingAt(i, str);
23
24
25
        public void insertSubstringStartingAt(int i, String str) {
26
          TrieNode node = root;
27
          for (int j = i; j < str.length(); j++) {</pre>
            char letter = str.charAt(j);
28
29
            if (!node.children.containsKey(letter)) {
30
              TrieNode newNode = new TrieNode();
              node.children.put(letter, newNode);
31
32
33
            node = node.children.get(letter);
```

```
Solution 1
             Solution 2
 1 import java.util.*;
 3
   class Program {
     // Do not edit the class below except for the
      // populateSuffixTrieFrom and contains methods.
     \ensuremath{//} Feel free to add new properties and methods
      // to the class.
      static class TrieNode {
9
       Map<Character, TrieNode> children = new HashMap<Character,</pre>
10
11
12
      static class SuffixTrie {
        TrieNode root = new TrieNode();
13
14
        char endSymbol = '*';
15
16
        public SuffixTrie(String str) {
17
         populateSuffixTrieFrom(str);
18
19
20
        public void populateSuffixTrieFrom(String str) {
21
         // Write your code here.
22
23
        public boolean contains(String str) {
24
25
         // Write your code here.
26
         return false;
27
28
29
30
```



Run or submit code when you're ready.