Our Solution(s) Run Code

Solution 1

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

import java.util.*;

class Program {
  static class TrieNode {
    Map<Character, TrieNode> children = new HashMap<Character,
  }

  static class SuffixTrie {
    TrieNode root = new TrieNode();
    char endSymbol = '*';

    public SuffixTrie(String str) {
      populateSuffixTrieFrom(str);
    }

    // O(n^2) time | O(n^2) space
    public void populateSuffixTrieFrom(String str) {
      for (int i = 0; i < str.length(); i++) {
        insertSubstringStartingAt(i, str);
      }
    }

    public void insertSubstringStartingAt(int i, String str) {
      TrieNode node = root;
      for (int j = i; j < str.length(); j++) {
        char letter = str.charAt(j);
        if (!node.children.containsKey(letter)) {
          TrieNode newNode = new TrieNode();
          node.children.put(letter, newNode);
        }
        node = node.children.get(letter);
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
import java.util.*;

class Program {
  // Do not edit the class below except for the
  // populateSuffixTrieFrom and contains methods.
  // Feel free to add new properties and methods
  // to the class.
  static class TrieNode {
    Map<Character, TrieNode> children = new HashMap<Character,
  }

  static class SuffixTrie {
    TrieNode root = new TrieNode();
    char endSymbol = '*';

    public SuffixTrie(String str) {
      populateSuffixTrieFrom(str);
    }

    public void populateSuffixTrieFrom(String str) {
      // Write your code here.
    }

    public boolean contains(String str) {
      // Write your code here.
      return false;
    }
  }
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.