

Prompt	Scratchpad	Our Solution(s)	Video Explanation	Run Code	Your Solutions	Run Code
--------	------------	-----------------	-------------------	----------	----------------	----------

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 using System.Collections.Generic;
4
5 public class Program {
6     public class TrieNode {
7         public Dictionary<char, TrieNode> Children = new Dictionary<char, TrieNode>()
8     }
9
10    public class SuffixTrie {
11        public TrieNode root = new TrieNode();
12        public char endSymbol = '*';
13
14        public SuffixTrie(string str) {
15            PopulateSuffixTrieFrom(str);
16        }
17
18        // O(n^2) time | O(n^2) space
19        public void PopulateSuffixTrieFrom(string str) {
20            for (int i = 0; i < str.Length; i++) {
21                insertSubstringStartingAt(i, str);
22            }
23        }
24
25        public void insertSubstringStartingAt(int i, string str) {
26            TrieNode node = root;
27            for (int j = i; j < str.Length; j++) {
28                char letter = str[j];
29                if (!node.Children.ContainsKey(letter)) {
30                    TrieNode newNode = new TrieNode();
31                    node.Children.Add(letter, newNode);
32                }
33                node = node.Children[letter];
34            }
35            node.Children[endSymbol] = null;
36        }
37
38        // O(m) time | O(1) space
39        public bool Contains(string str) {
40            TrieNode node = root;
41            for (int i = 0; i < str.Length; i++) {
42                char letter = str[i];
43                if (!node.Children.ContainsKey(letter)) {
44                    return false;
45                }
46                node = node.Children[letter];
47            }
48            return node.Children.ContainsKey(endSymbol);
49        }
50    }
51 }
52
```

Solution 1 Solution 2 Solution 3

```
1 using System.Collections.Generic;
2
3 public class Program {
4     // Do not edit the class below except for the
5     // PopulateSuffixTrieFrom and Contains methods.
6     // Feel free to add new properties and methods
7     // to the class.
8     public class TrieNode {
9         public Dictionary<char, TrieNode> Children = new Dictionary<char, TrieNode>()
10    }
11
12    public class SuffixTrie {
13        public TrieNode root = new TrieNode();
14        public 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
24        public bool Contains(string str) {
25            // Write your code here.
26            return false;
27        }
28    }
29 }
30
```

Run or submit code when you're ready.