

PromptScratchpadOur Solution(s)Video ExplanationRun Code

Solution 1

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

Your SolutionsRun Code

Solution 1Solution 2Solution 3

```
1 class Program {
2     // Do not edit the class below except for the
3     // populateSuffixTrieFrom and contains methods.
4     // Feel free to add new properties and methods
5     // to the class.
6     class TrieNode {
7         var children: [String: Any] = [:]
8     }
9
10    class SuffixTrie {
11        var root = TrieNode()
12        let endSymbol = "*"
13
14        init(string: String) {
15            populateSuffixTrieFrom(string: string)
16        }
17
18        func populateSuffixTrieFrom(string: String) {
19            // Write your code here.
20        }
21
22        func contains(string: String) -> Bool {
23            // Write your code here.
24            return false
25        }
26    }
27 }
28
```

Custom OutputRaw OutputSubmit Code

Run or submit code when you're ready.