

Our Solution(s)

Run Code

Solution 1

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class SuffixTrie:
4     def __init__(self, string):
5         self.root = {}
6         self.endSymbol = "*"
7         self.populateSuffixTrieFrom(string)
8
9     # O(n^2) time | O(n^2) space
10    def populateSuffixTrieFrom(self, string):
11        for i in range(len(string)):
12            self.insertSubstringStartingAt(i, string)
13
14    def insertSubstringStartingAt(self, i, string):
15        node = self.root
16        for j in range(i, len(string)):
17            letter = string[j]
18            if letter not in node:
19                node[letter] = {}
20            node = node[letter]
21        node[self.endSymbol] = True
22
23    # O(m) time | O(1) space
24    def contains(self, string):
25        node = self.root
26        for letter in string:
27            if letter not in node:
28                return False
29            node = node[letter]
30        return self.endSymbol in node
31
```

Our Tests

Your Solutions

Run Code

Solution 1   Solution 2   Solution 3

```
1 # Do not edit the class below except for the
2 # populateSuffixTrieFrom and contains methods.
3 # Feel free to add new properties and methods
4 # to the class.
5 class SuffixTrie:
6     def __init__(self, string):
7         self.root = {}
8         self.endSymbol = "*"
9         self.populateSuffixTrieFrom(string)
10
11    def populateSuffixTrieFrom(self, string):
12        # Write your code here.
13        pass
14
15    def contains(self, string):
16        # Write your code here.
17        pass
18
```

Custom Output

Submit Code

Run or submit code when you're ready.

Run or submit code when you're ready.