Our Solution(s)   Run Code

Solution 1

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class SuffixTrie:
    def __init__(self, string):
        self.root = {}
        self.endSymbol = "*"
        self.populateSuffixTrieFrom(string)

    # O(n^2) time | O(n^2) space
    def populateSuffixTrieFrom(self, string):
        for i in range(len(string)):
            self.insertSubstringStartingAt(i, string)

    def insertSubstringStartingAt(self, i, string):
        node = self.root
        for j in range(i, len(string)):
            letter = string[j]
            if letter not in node:
                node[letter] = {}
            node = node[letter]
        node[self.endSymbol] = True

    # O(m) time | O(1) space
    def contains(self, string):
        node = self.root
        for letter in string:
            if letter not in node:
                return False
            node = node[letter]
        return self.endSymbol in node
```

Your Solutions   Run Code

Solution 1   Solution 2   Solution 3

```python
# Do not edit the class below except for the
# populateSuffixTrieFrom and contains methods.
# Feel free to add new properties and methods
# to the class.
class SuffixTrie:
    def __init__(self, string):
        self.root = {}
        self.endSymbol = "*"
        self.populateSuffixTrieFrom(string)

    def populateSuffixTrieFrom(self, string):
        # Write your code here.
        pass

    def contains(self, string):
        # Write your code here.
        pass
```

Custom Output   Raw Output   Submit Code

```python
    def populateSuffixTrieFrom(self, string):
```

Run or submit code when you're ready.