

Our Solution(s)

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class SuffixTrie {
4   constructor(string) {
5     this.root = {};
6     this.endSymbol = '*';
7     this.populateSuffixTrieFrom(string);
8   }
9
10  // O(n^2) time | O(n^2) space
11  populateSuffixTrieFrom(string) {
12    for (let i = 0; i < string.length; i++) {
13      this.insertSubstringStartingAt(i, string);
14    }
15  }
16
17  insertSubstringStartingAt(i, string) {
18    let node = this.root;
19    for (let j = i; j < string.length; j++) {
20      const letter = string[j];
21      if (!(letter in node)) node[letter] = {};
22      node = node[letter];
23    }
24    node[this.endSymbol] = true;
25  }
26
27  // O(m) time | O(1) space
28  contains(string) {
29    let node = this.root;
30    for (const letter of string) {
31      if (!(letter in node)) return false;
32      node = node[letter];
33    }
34  }
```

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   constructor(string) {
7     this.root = {};
8     this.endSymbol = '*';
9     this.populateSuffixTrieFrom(string);
10  }
11
12  populateSuffixTrieFrom(string) {
13    // Write your code here.
14  }
15
16  contains(string) {
17    // Write your code here.
18  }
19 }
20
21 // Do not edit the line below.
22 exports.SuffixTrie = SuffixTrie;
23
```

