AlgoExpert Quad Layout Go 14px Sublime 00:00:00

Our Solution(s) Run Code

Solution 1

```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

type SuffixTrie map[byte]SuffixTrie

func NewSuffixTrie() SuffixTrie {
  trie := SuffixTrie{}
  return trie
}

// O(n^2) time | O(n^2) space
func (trie SuffixTrie) populateSuffixTrieFrom(str string) {
  for i := range str {
    node := trie
    for j := i; j < len(str); j++ {
      letter := str[j]
      if _, found := node[letter]; !found {
        node[letter] = NewSuffixTrie()
      }
      node = node[letter]
    }
    node['*'] = nil
  }
}

// O(m) time | O(1) space
func (trie SuffixTrie) contains(str string) bool {
  node := trie
  for i := 0; i < len(str); i++ {
    letter := str[i]
    if _, found := node[letter]; !found {
      return false
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

// Do not edit the class below except for the
// populateSuffixTrieFrom and contains methods.
// Feel free to add new properties and methods
// to the class.
type SuffixTrie map[byte]SuffixTrie

func (trie SuffixTrie) populateSuffixTrieFrom(str string) {
  // Write your code here.
}

func (trie SuffixTrie) contains(str string) bool {
  // Write your code here.
  return false
}
```

Our Tests

Custom Output Submit Code

```go
package main
```

```go
func (trie suffixTrie) Equals(other suffixTrie) bool {
  if len(trie) != len(other) {
    return false
  }
  for key, child := range trie {
    otherChild, found := other[key]
    if !found {
      return false
    } else if child != nil && !child.Equals(otherChild) {
      return false
    }
  }
  return true
}

func trieFromString(str string) suffixTrie {
  trie := suffixTrie{}
  trie.populateSuffixTrieFrom(str)
  return trie
}
```

Run or submit code when you're ready.