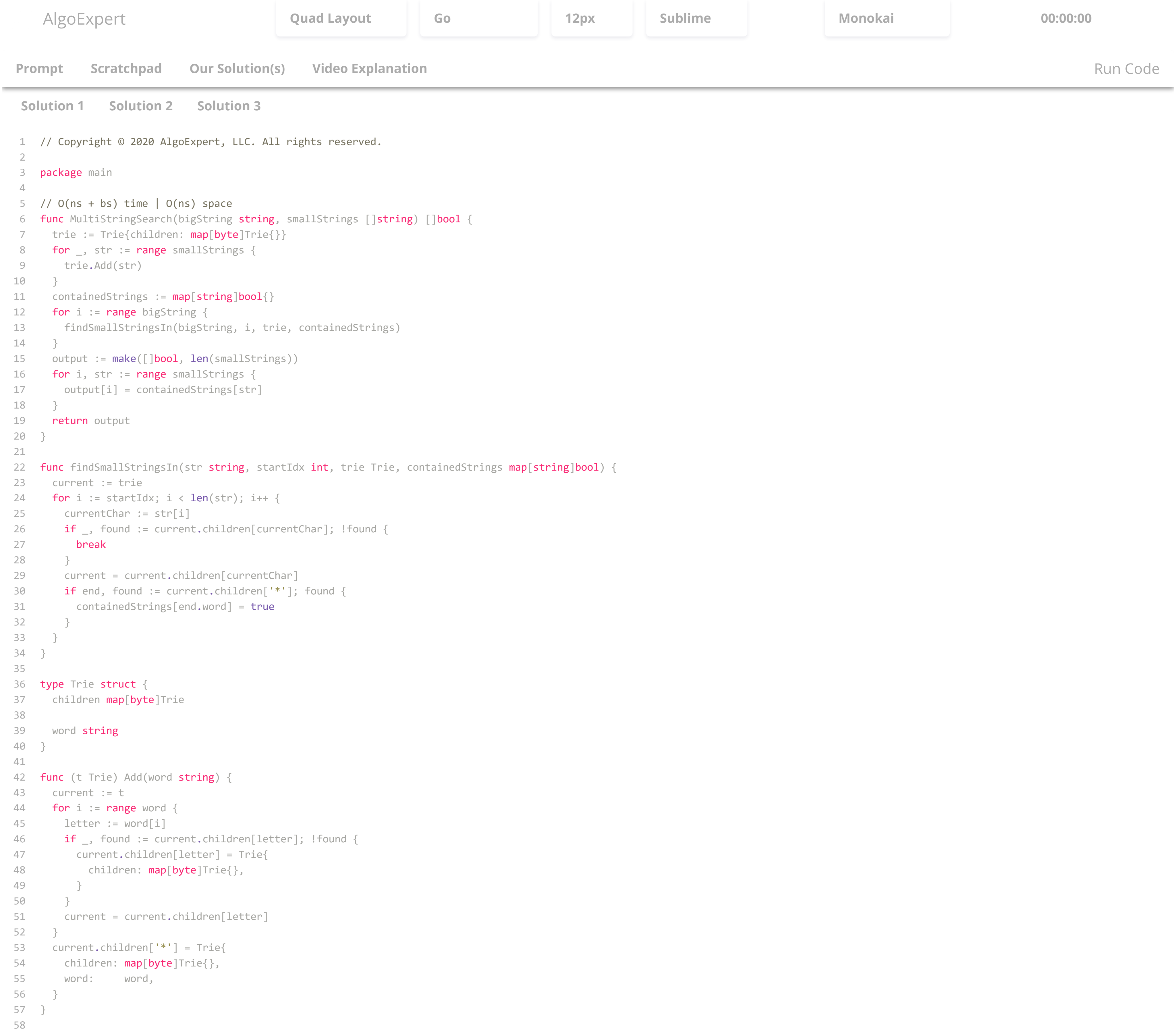

```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

// O(ns + bs) time | O(ns) space
func MultiStringSearch(bigString string, smallStrings []string) []bool {
  trie := Trie{children: map[byte]Trie{}}
  for _, str := range smallStrings {
    trie.Add(str)
  }
  containedStrings := map[string]bool{}
  for i := range bigString {
    findSmallStringsIn(bigString, i, trie, containedStrings)
  }
  output := make([]bool, len(smallStrings))
  for i, str := range smallStrings {
    output[i] = containedStrings[str]
  }
  return output
}

func findSmallStringsIn(str string, startIdx int, trie Trie, containedStrings map[string]bool) {
  current := trie
  for i := startIdx; i < len(str); i++ {
    currentChar := str[i]
    if _, found := current.children[currentChar]; !found {
      break
    }
    current = current.children[currentChar]
    if end, found := current.children['*']; found {
      containedStrings[end.word] = true
    }
  }
}

type Trie struct {
  children map[byte]Trie

  word string
}

func (t Trie) Add(word string) {
  current := t
  for i := range word {
    letter := word[i]
    if _, found := current.children[letter]; !found {
      current.children[letter] = Trie{
        children: map[byte]Trie{},
      }
    }
    current = current.children[letter]
  }
  current.children['*'] = Trie{
    children: map[byte]Trie{},
    word:     word,
  }
}
```