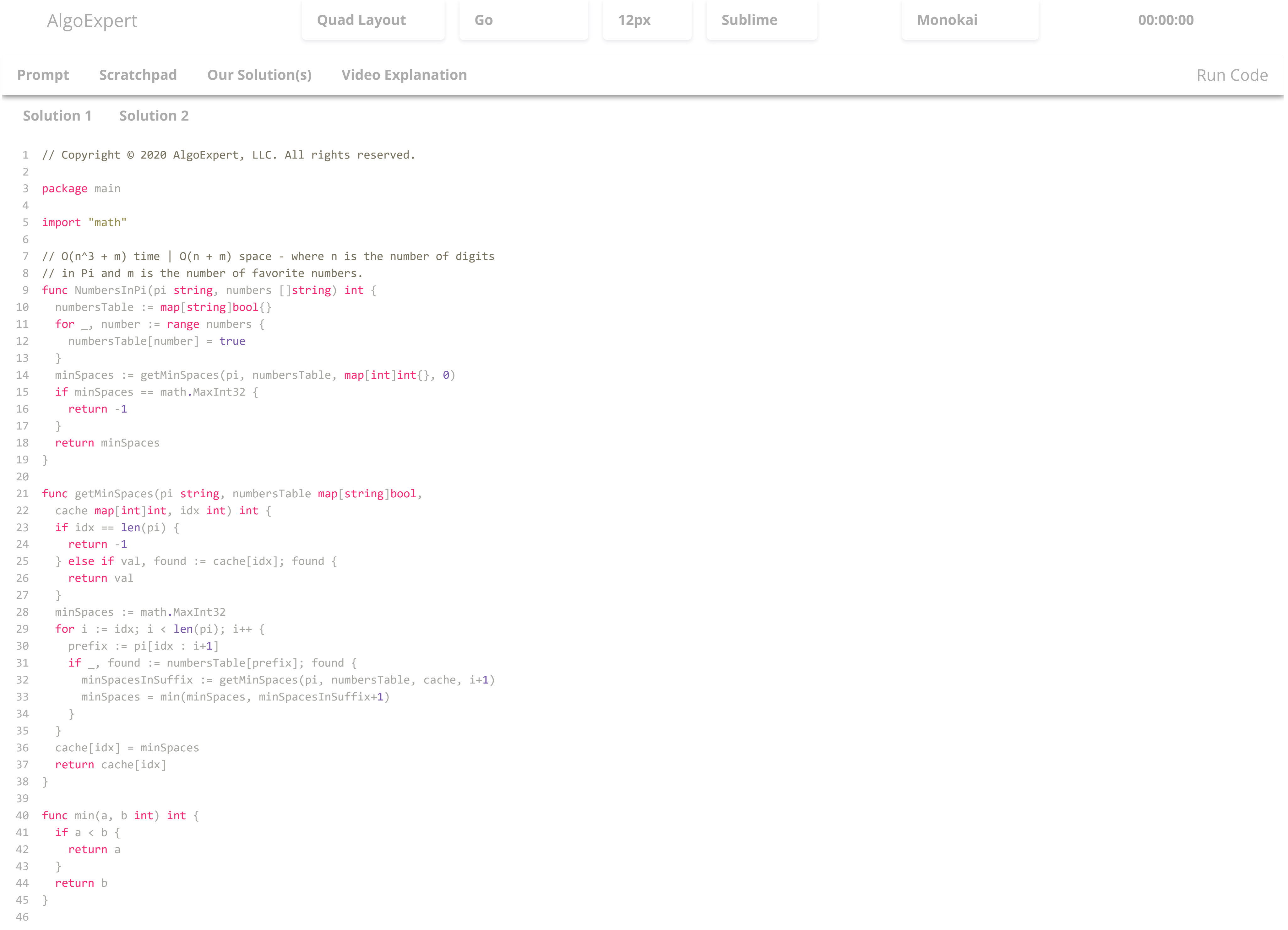

```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

import "math"

// O(n^3 + m) time | O(n + m) space - where n is the number of digits
// in Pi and m is the number of favorite numbers.
func NumbersInPi(pi string, numbers []string) int {
  numbersTable := map[string]bool{}
  for _, number := range numbers {
    numbersTable[number] = true
  }
  minSpaces := getMinSpaces(pi, numbersTable, map[int]int{}, 0)
  if minSpaces == math.MaxInt32 {
    return -1
  }
  return minSpaces
}

func getMinSpaces(pi string, numbersTable map[string]bool,
  cache map[int]int, idx int) int {
  if idx == len(pi) {
    return -1
  } else if val, found := cache[idx]; found {
    return val
  }
  minSpaces := math.MaxInt32
  for i := idx; i < len(pi); i++ {
    prefix := pi[idx : i+1]
    if _, found := numbersTable[prefix]; found {
      minSpacesInSuffix := getMinSpaces(pi, numbersTable, cache, i+1)
      minSpaces = min(minSpaces, minSpacesInSuffix+1)
    }
  }
  cache[idx] = minSpaces
  return cache[idx]
}

func min(a, b int) int {
  if a < b {
    return a
  }
  return b
}
```