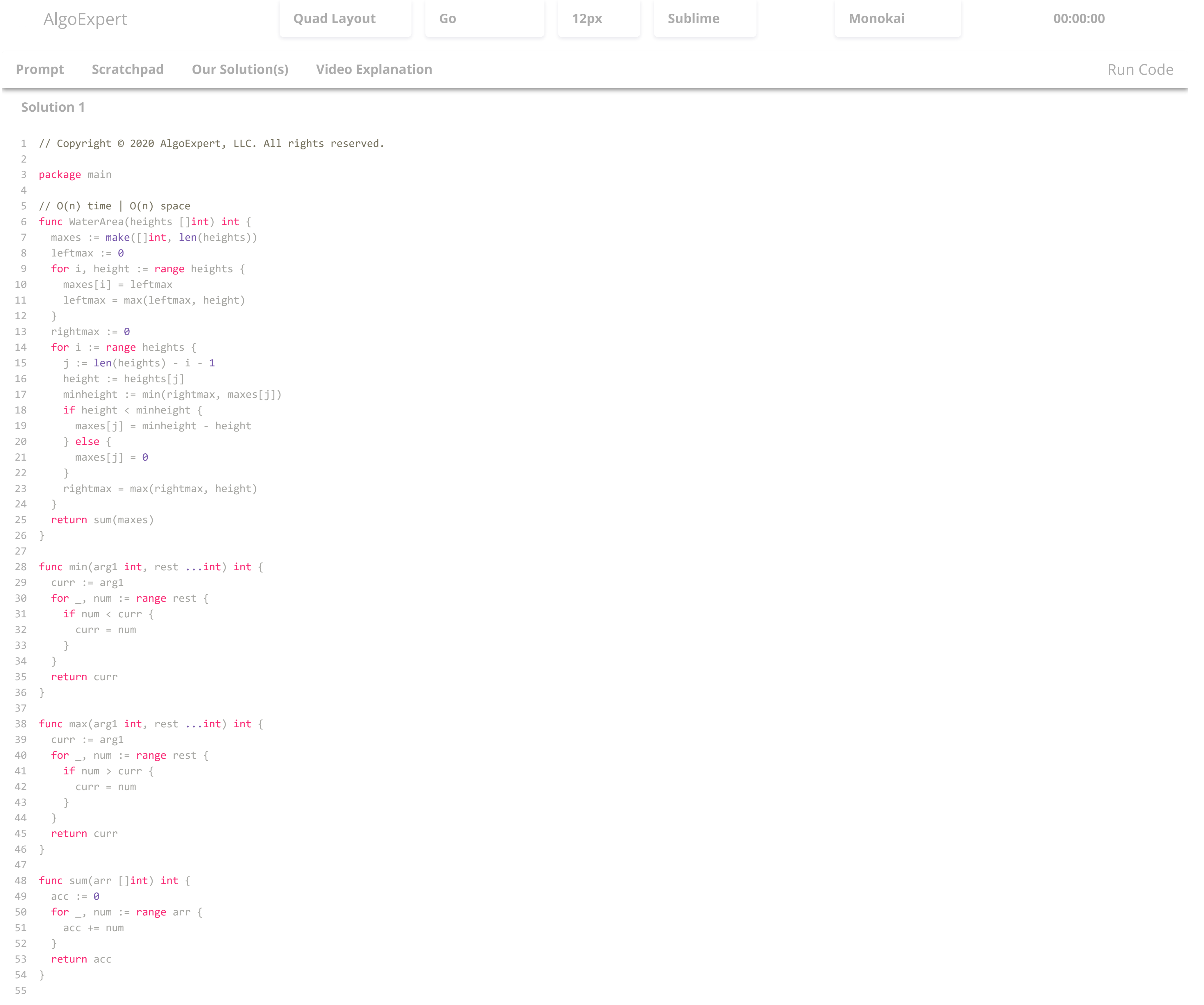

Solution 1

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

package main

// O(n) time | O(n) space
func WaterArea(heights []int) int {
  maxes := make([]int, len(heights))
  leftmax := 0
  for i, height := range heights {
    maxes[i] = leftmax
    leftmax = max(leftmax, height)
  }
  rightmax := 0
  for i := range heights {
    j := len(heights) - i - 1
    height := heights[j]
    minheight := min(rightmax, maxes[j])
    if height < minheight {
      maxes[j] = minheight - height
    } else {
      maxes[j] = 0
    }
    rightmax = max(rightmax, height)
  }
  return sum(maxes)
}

func min(arg1 int, rest ...int) int {
  curr := arg1
  for _, num := range rest {
    if num < curr {
      curr = num
    }
  }
  return curr
}

func max(arg1 int, rest ...int) int {
  curr := arg1
  for _, num := range rest {
    if num > curr {
      curr = num
    }
  }
  return curr
}

func sum(arr []int) int {
  acc := 0
  for _, num := range arr {
    acc += num
  }
  return acc
}

```