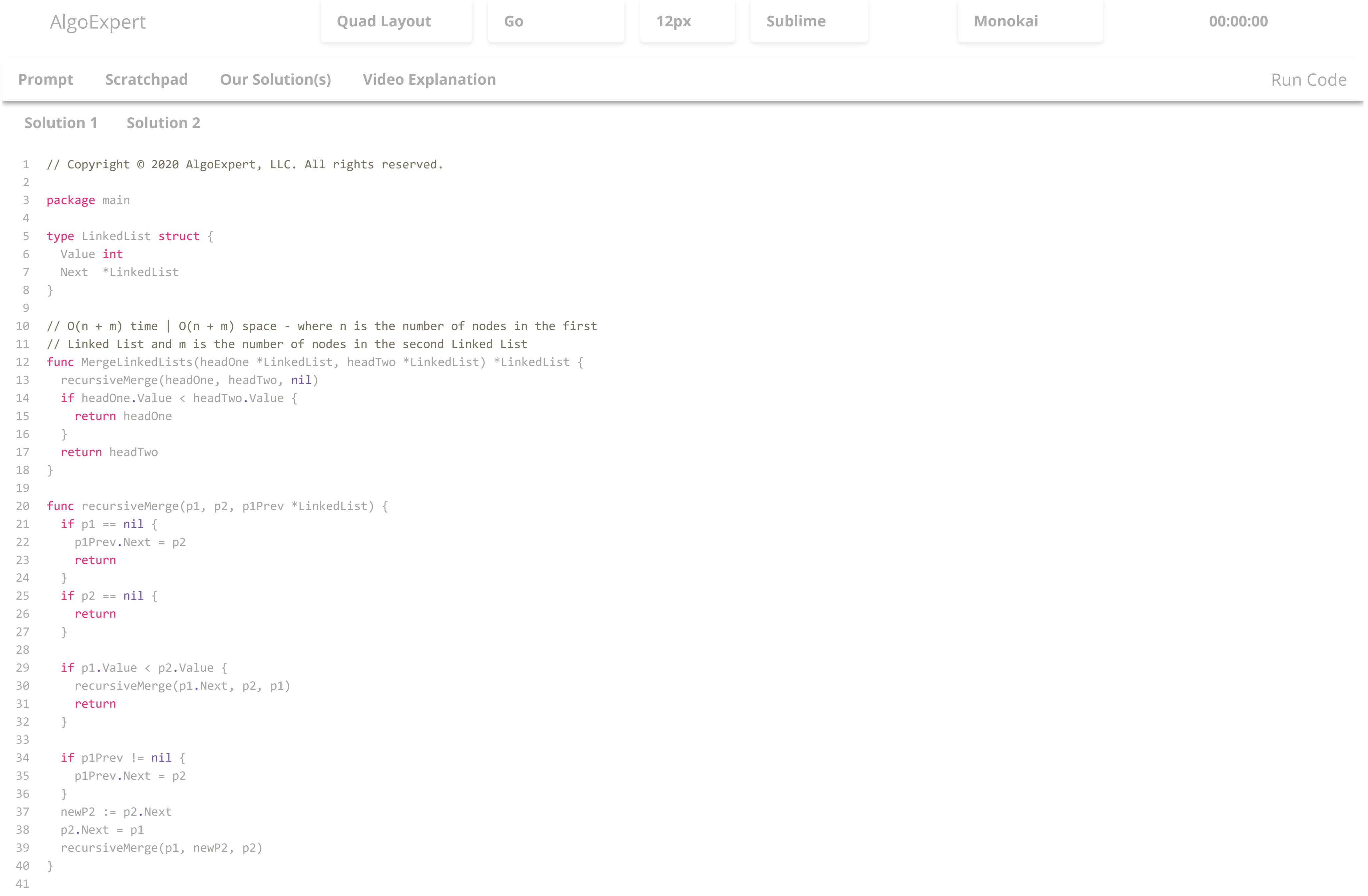

```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

type LinkedList struct {
  Value int
  Next  *LinkedList
}

// O(n + m) time | O(n + m) space - where n is the number of nodes in the first
// Linked List and m is the number of nodes in the second Linked List
func MergeLinkedLists(headOne *LinkedList, headTwo *LinkedList) *LinkedList {
  recursiveMerge(headOne, headTwo, nil)
  if headOne.Value < headTwo.Value {
    return headOne
  }
  return headTwo
}

func recursiveMerge(p1, p2, p1Prev *LinkedList) {
  if p1 == nil {
    p1Prev.Next = p2
    return
  }
  if p2 == nil {
    return
  }

  if p1.Value < p2.Value {
    recursiveMerge(p1.Next, p2, p1)
    return
  }

  if p1Prev != nil {
    p1Prev.Next = p2
  }
  newP2 := p2.Next
  p2.Next = p1
  recursiveMerge(p1, newP2, p2)
}

```