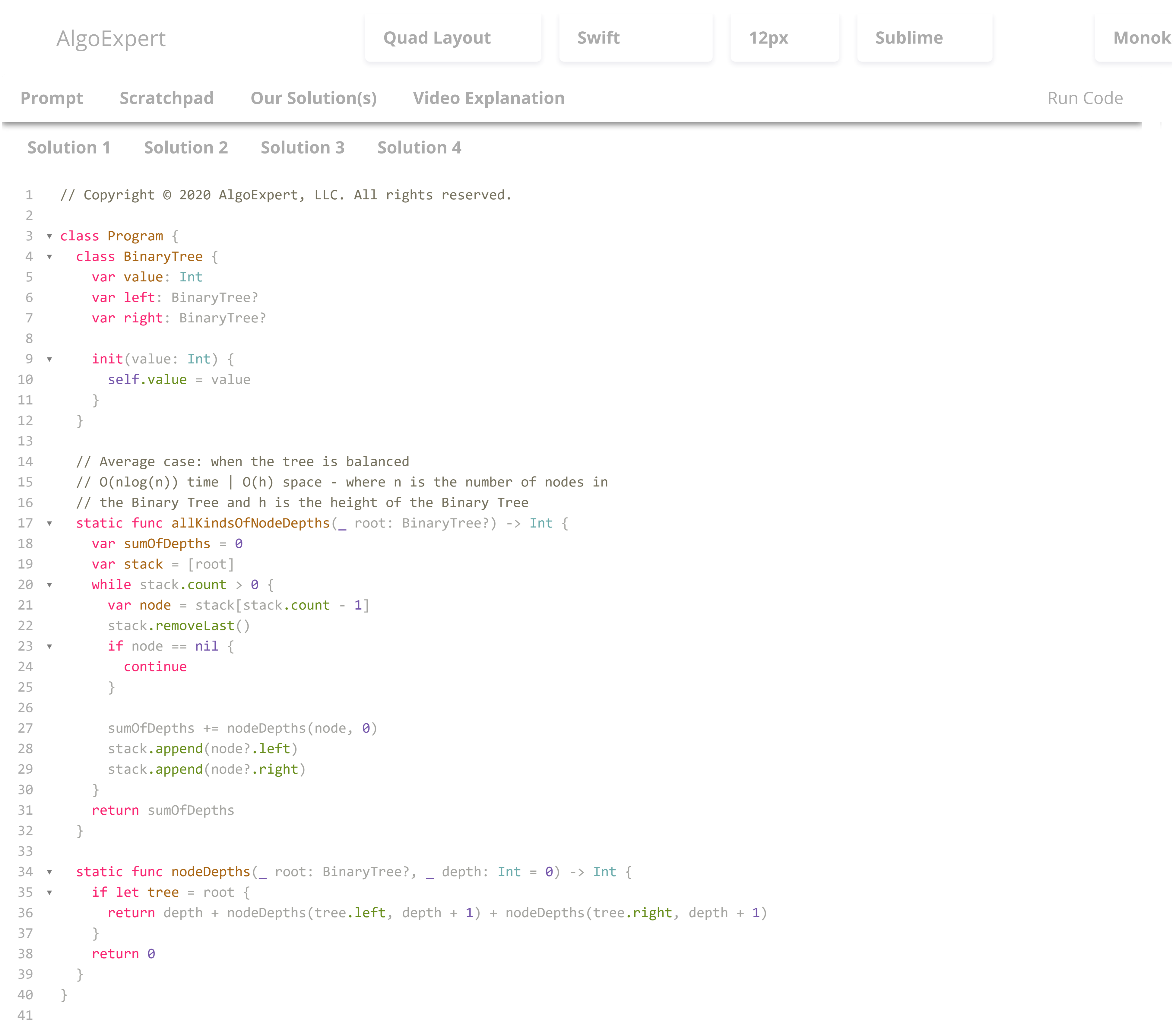

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
  class BinaryTree {
    var value: Int
    var left: BinaryTree?
    var right: BinaryTree?

    init(value: Int) {
      self.value = value
    }
  }

  // Average case: when the tree is balanced
  // O(nlog(n)) time | O(h) space - where n is the number of nodes in
  // the Binary Tree and h is the height of the Binary Tree
  static func allKindsOfNodeDepths(_ root: BinaryTree?) -> Int {
    var sumOfDepths = 0
    var stack = [root]
    while stack.count > 0 {
      var node = stack[stack.count - 1]
      stack.removeLast()
      if node == nil {
        continue
      }

      sumOfDepths += nodeDepths(node, 0)
      stack.append(node?.left)
      stack.append(node?.right)
    }
    return sumOfDepths
  }

  static func nodeDepths(_ root: BinaryTree?, _ depth: Int = 0) -> Int {
    if let tree = root {
      return depth + nodeDepths(tree.left, depth + 1) + nodeDepths(tree.right, depth + 1)
    }
    return 0
  }
}
```