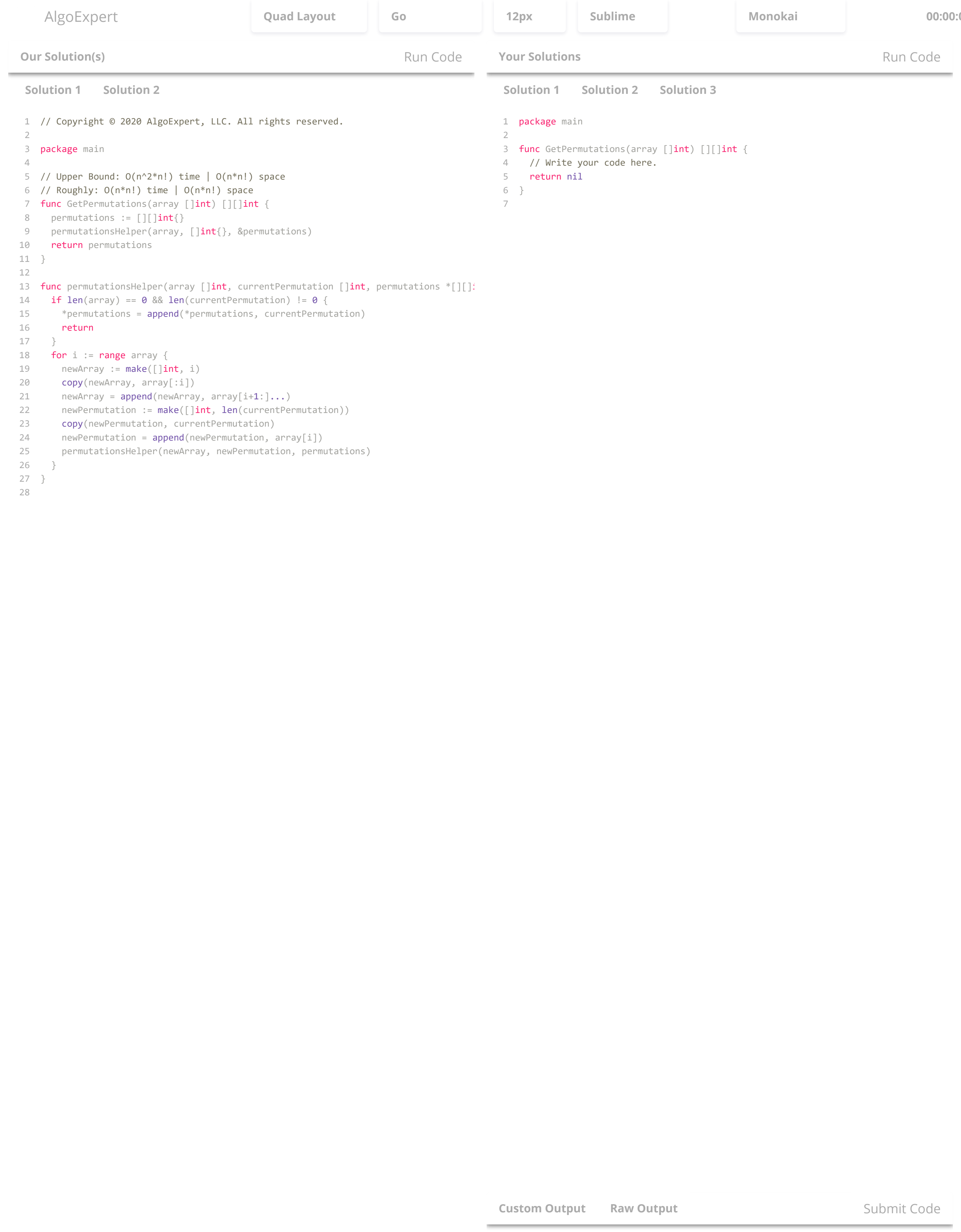

```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

// Upper Bound: O(n^2*n!) time | O(n*n!) space
// Roughly: O(n*n!) time | O(n*n!) space
func GetPermutations(array []int) [][]int {
  permutations := [][]int{}
  permutationsHelper(array, []int{}, &permutations)
  return permutations
}

func permutationsHelper(array []int, currentPermutation []int, permutations *[][]
  if len(array) == 0 && len(currentPermutation) != 0 {
    *permutations = append(*permutations, currentPermutation)
    return
  }
  for i := range array {
    newArray := make([]int, i)
    copy(newArray, array[:i])
    newArray = append(newArray, array[i+1:]...)
    newPermutation := make([]int, len(currentPermutation))
    copy(newPermutation, currentPermutation)
    newPermutation = append(newPermutation, array[i])
    permutationsHelper(newArray, newPermutation, permutations)
  }
}

```

```go
package main

func GetPermutations(array []int) [][]int {
  // Write your code here.
  return nil
}

```

Run or submit code when you're ready.