Solution 1

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

package main

// Average: O(n^2) time | O(n^2) space
// Worst: O(n^3) time | O(n^2) space
func FourNumberSum(array []int, target int) [][]int {
  allPairSums := map[int][][]int{}
  quadruplets := [][]int{}
  for i := 1; i < len(array)-1; i++ {
    for j := i + 1; j < len(array); j++ {
      currentSum := array[i] + array[j]
      difference := target - currentSum
      if pairs, found := allPairSums[difference]; found {
        for _, pair := range pairs {
          newquad := append(pair, array[i], array[j])
          quadruplets = append(quadruplets, newquad)
        }
      }
    }
    for k := 0; k < i; k++ {
      currentSum := array[i] + array[k]
      allPairSums[currentSum] = append(allPairSums[currentSum], []int{array[k], array[i]})
    }
  }
  return quadruplets
}

```