Solution 1

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

using System.Collections.Generic;

// Average: O(n^2) time | O(n^2) space
// Worst: O(n^3) time | O(n^2) space
public class Program {
  public static List<int[]> FourNumberSum(int[] array, int targetSum) {
    Dictionary<int, List<int[]> > allPairSums = new Dictionary<int, List<int[]> >();
    List<int[]> quadruplets = new List<int[]>();
    for (int i = 1; i < array.Length - 1; i++) {
      for (int j = i + 1; j < array.Length; j++) {
        int currentSum = array[i] + array[j];
        int difference = targetSum - currentSum;
        if (allPairSums.ContainsKey(difference)) {
          foreach (int[] pair in allPairSums[difference]) {
            int[] newQuadruplet =
            {pair[0], pair[1], array[i], array[j]};
            quadruplets.Add(newQuadruplet);
          }
        }
      }
      for (int k = 0; k < i; k++) {
        int currentSum = array[i] + array[k];
        int[] pair = {array[k], array[i]};
        if (!allPairSums.ContainsKey(currentSum)) {
          List<int[]> pairGroup = new List<int[]>();
          pairGroup.Add(pair);
          allPairSums.Add(currentSum, pairGroup);
        } else {
          allPairSums[currentSum].Add(pair);
        }
      }
    }
    return quadruplets;
  }
}

```