Our Solution(s) Run Code

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, L
    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);
        }
      }
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
using System.Collections.Generic;

public class Program {
  public static List<int[]> FourNumberSum(int[] array, int targetSum)
    // Write your code here.
    return null;
  }
}

```

Our Tests

Custom Output Submit Code

```
        Array.Sort(quad1);
      }
      foreach (int[] quad in quad2) {
        Array.Sort(quad);
      }
      foreach (int[] quad1 in quad1) {
        bool found = false;
        foreach (int[] quad2 in quad2) {
          if (Enumerable.SequenceEqual(quad1, quad2)) {
            found = true;
            break;
          }
        }
        if (found == false) {
          return false;
```

Run or submit code when you're ready.