Our Solution(s) Run Code

Solution 1

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

using System;
using System.Collections.Generic;

public class Program {
  // O(n^2) time | O(n) space
  public static List<int[]> DiskStacking(List<int[]> disks) {
    disks.Sort((disk1, disk2) => disk1[2].CompareTo(disk2[2]));
    int[] heights = new int[disks.Count];
    for (int i = 0; i < disks.Count; i++) {
      heights[i] = disks[i][2];
    }
    int[] sequences = new int[disks.Count];
    for (int i = 0; i < disks.Count; i++) {
      sequences[i] = Int32.MinValue;
    }
    int maxHeightIdx = 0;
    for (int i = 1; i < disks.Count; i++) {
      int[] currentDisk = disks[i];
      for (int j = 0; j < i; j++) {
        int[] otherDisk = disks[j];
        if (areValidDimensions(otherDisk, currentDisk)) {
          if (heights[i] <= currentDisk[2] + heights[j]) {
            heights[i] = currentDisk[2] + heights[j];
            sequences[i] = j;
          }
        }
      }
      if (heights[i] >= heights[maxHeightIdx]) {
        maxHeightIdx = i;
      }
    }
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
using System.Collections.Generic;

public class Program {
  public static List<int[]> DiskStacking(List<int[]> disks) {
    // Write your code here.
    return null;
  }
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.