Our Solution(s)

```
Run Code
```

```
Your Solutions
```

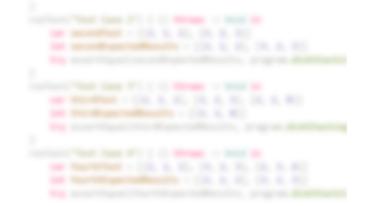
14рх

```
Run Code
```

```
Solution 1
 1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
   class Program {
        // O(n^2) time | O(1) space
 4
        func diskStacking(disks: inout [[Int]]) -> [[Int]] {
            disks.sort(by: { $0[2] < $1[2] })
            var heights = disks.map { $0[2] }
            var previousIndices = Array(repeating: -1, count: disks.count
 9
10
            var maximumHeightIndex = 0
11
12
            for i in 1 ..< disks.count {</pre>
13
                let currentDisk = disks[i]
14
                for j in 0 ..< i {
16
                    let previousDisk = disks[j]
17
                    if areValidDimensions(previousDisk, currentDisk) {
18
19
                        if heights[i] <= heights[j] + currentDisk[2] {</pre>
20
                            heights[i] = heights[j] + currentDisk[2]
21
                            previousIndices[i] = j
22
23
                }
25
26
                if heights[i] >= heights[maximumHeightIndex] {
27
                    maximumHeightIndex = i
28
29
30
31
            return buildSequence(disks, previousIndices, &maximumHeightInd
32
33
```

```
class Program {
   func diskStacking(disks: inout [[Int]]) -> [[Int]] {
      // Write your code here.
      return []
   }
}
```

and the last of the same of th



Run or submit code when you're ready.