33

14px

**Your Solutions** 

Run Code

Our Solution(s) Run Code

```
Solution 1
 1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
   class Program {
        // O(nc) time | O(nc) space
        func knapsackProblem(_ items: [[Int]], _ capacity: Int) -> (Int,
            var knapsackValues = [[Int]]()
            for _ in 0 ..< items.count + 1 {</pre>
 9
                let row = Array(repeating: 0, count: capacity + 1)
10
                knapsackValues.append(row)
11
12
            for currentItemIndex in 1 ... items.count + 1 {
14
                let currentValue = items[currentItemIndex - 1][0]
                let currentWeight = items[currentItemIndex - 1][1]
16
17
                for currentCapacity in 0 ..< capacity + 1 {</pre>
                    if currentWeight <= currentCapacity {</pre>
18
19
                        knapsackValues[currentItemIndex][currentCapacity]
20
21
                        knapsackValues[currentItemIndex][currentCapacity]
26
            return (knapsackValues[items.count][capacity], getKnapsackItem
27
28
29
        func getKnapsackItems(_ items: [[Int]], _ knapsackValues: [[Int]])
30
            var sequence = [Int]()
31
            var currentItemIndex = knapsackValues.count - 1
```

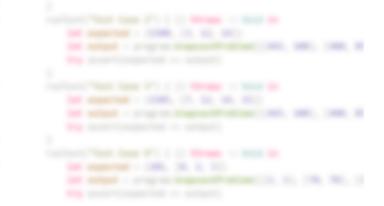
var currentCapacity = knapsackValues[0].count - 1

\_\_\_\_

and the last of th

\_\_\_\_

 Our Tests
 Custom Output
 Submit Code



Run or submit code when you're ready.