11

Run Code

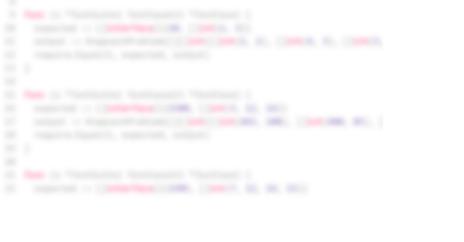
Our Solution(s)

Solution 1

Run Code

```
Your Solutions
```

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
3
   package main
   func KnapsackProblem(items [][]int, capacity int) []interface{} {
     values := make([][]int, len(items)+1)
     for i := range values {
       values[i] = make([]int, capacity+1)
9
     for i := 1; i < len(items)+1; i++ {</pre>
10
11
       currentValue := items[i-1][0]
12
       currentWeight := items[i-1][1]
13
        for c := 0; c < capacity+1; c++ {</pre>
         if currentWeight > c {
14
15
           values[i][c] = values[i-1][c]
16
         } else {
17
            values[i][c] = max(values[i-1][c], values[i-1][c-currentWeight
18
19
20
21
22
     value := values[len(items)][capacity]
23
     sequence := getKnapsackItems(values, items)
24
     return []interface{}{value, sequence}
25 }
26
27
   func getKnapsackItems(values [][]int, items [][]int) []int {
28
     sequence := []int{}
29
     i, c := len(values)-1, len(values[0])-1
30
     for i > 0 {
31
       if values[i][c] == values[i-1][c] {
32
        i--
33
       } else {
```



Run or submit code when you're ready.