Quad Layout Sublime Monokai AlgoExpert Go 12px 00:00:00

Prompt Run Code

58 59

60 } 61

return curr

```
Scratchpad
  Our Solution(s)
```

Video Explanation

Solution 1

 $1\,$ // Copyright @ 2020 AlgoExpert, LLC. All rights reserved. package main 5 func KnapsackProblem(items [][]int, capacity int) []interface{} { values := make([][]int, len(items)+1) for i := range values { 8 values[i] = make([]int, capacity+1) 9 10 for i := 1; i < len(items)+1; i++ {</pre> 11 currentValue := items[i-1][0] 12 currentWeight := items[i-1][1] 13 for c := 0; c < capacity+1; c++ {</pre> 14 if currentWeight > c { 15 values[i][c] = values[i-1][c] 16 values[i][c] = max(values[i-1][c], values[i-1][c-currentWeight]+currentValue) 17 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])-130 for i > 0 { 31 if values[i][c] == values[i-1][c] { 32 } else { 33 34 sequence = append(sequence, i-1) 35 c -= items[i-1][1] 36 i--37 **if** c == 0 { 38 39 break 40 41 reverse(sequence) 43 return sequence 44 45 func reverse(numbers []int) { 46 for i, j := 0, len(numbers)-1; i < j; i, j = i+1, j-1 {</pre> 47 48 numbers[i], numbers[j] = numbers[j], numbers[i] 49 50 51 52 func max(arg int, rest ...int) int { 53 curr := arg 54 for $_$, num := range rest { 55 if curr < num {</pre> 56 curr = num 57