

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.*;
4
5 class Program {
6     // O(nc) time | O(nc) space
7     public static List<List<Integer>> knapsackProblem(int[][] items, int capacity) {
8         int[][] knapsackValues = new int[items.length + 1][capacity + 1];
9         for (int i = 1; i < items.length + 1; i++) {
10             int currentWeight = items[i - 1][1];
11             int currentValue = items[i - 1][0];
12             for (int c = 0; c < capacity + 1; c++) {
13                 if (currentWeight > c) {
14                     knapsackValues[i][c] = knapsackValues[i - 1][c];
15                 } else {
16                     knapsackValues[i][c] =
17                         Math.max(
18                             knapsackValues[i - 1][c],
19                             knapsackValues[i - 1][c - currentWeight] + currentValue
20                         )
21                 }
22             }
23         }
24         return getKnapsackItems(knapsackValues, items, knapsackValues[items.length][0]);
25     }
26
27     public static List<List<Integer>> getKnapsackItems(
28         int[][] knapsackValues, int[][] items, int weight) {
29         List<List<Integer>> sequence = new ArrayList<List<Integer>>();
30         List<Integer> totalWeight = new ArrayList<Integer>();
31         totalWeight.add(weight);
32         sequence.add(totalWeight);
33         for (int i = 0; i < items.length; i++) {
34             int currentWeight = items[i][1];
35             int currentValue = items[i][0];
36             List<List<Integer>> nextSequence = new ArrayList<List<Integer>>();
37             for (List<Integer> totalWeight : sequence) {
38                 List<Integer> nextTotalWeight = new ArrayList<Integer>();
39                 nextTotalWeight.add(totalWeight.get(0) + currentWeight);
40                 nextSequence.add(nextTotalWeight);
41             }
42             sequence = nextSequence;
43         }
44         return sequence;
45     }
46 }
```

Solution 1

Solution 2

Solution 3

```
1 import java.util.*;
2
3 class Program {
4     public static List<List<Integer>> knapsackProblem(int[][] items, int capacity) {
5         // Write your code here.
6         // Replace the code below.
7         List<Integer> totalValue = Arrays.asList(10);
8         List<Integer> finalItems = Arrays.asList(1, 2);
9         var result = new ArrayList<List<Integer>>();
10        result.add(totalValue);
11        result.add(finalItems);
12        return result;
13    }
14 }
15
```

Our Tests

Custom Output

Submit Code

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

2

3 import java.util.\*;

4

5 class Program {

6 // O(nc) time | O(nc) space

7 public static List<List<Integer>> knapsackProblem(int[][] items, int capacity) {

8 int[][] knapsackValues = new int[items.length + 1][capacity + 1];

9 for (int i = 1; i < items.length + 1; i++) {

10 int currentWeight = items[i - 1][1];

11 int currentValue = items[i - 1][0];

12 for (int c = 0; c < capacity + 1; c++) {

13 if (currentWeight > c) {

14 knapsackValues[i][c] = knapsackValues[i - 1][c];

15 } else {

16 knapsackValues[i][c] =

17 Math.max(

18 knapsackValues[i - 1][c],

19 knapsackValues[i - 1][c - currentWeight] + currentValue

20 )

21 }

22 }

23 }

24 return getKnapsackItems(knapsackValues, items, knapsackValues[items.length][0]);

25 }

26

27 public static List<List<Integer>> getKnapsackItems(

28 int[][] knapsackValues, int[][] items, int weight) {

29 List<List<Integer>> sequence = new ArrayList<List<Integer>>();

30 List<Integer> totalWeight = new ArrayList<Integer>();

31 totalWeight.add(weight);

32 sequence.add(totalWeight);

33 for (int i = 0; i < items.length; i++) {

34 int currentWeight = items[i][1];

35 int currentValue = items[i][0];

36 List<List<Integer>> nextSequence = new ArrayList<List<Integer>>();

37 for (List<Integer> totalWeight : sequence) {

38 List<Integer> nextTotalWeight = new ArrayList<Integer>();

39 nextTotalWeight.add(totalWeight.get(0) + currentWeight);

40 nextSequence.add(nextTotalWeight);

41 }

42 sequence = nextSequence;

43 }

44 return sequence;

45 }

46 }

1 import java.util.\*;

2

3 class Program {

4 public static List<List<Integer>> knapsackProblem(int[][] items, int capacity) {

5 // Write your code here.

6 // Replace the code below.

7 List<Integer> totalValue = Arrays.asList(10);

8 List<Integer> finalItems = Arrays.asList(1, 2);

9 var result = new ArrayList<List<Integer>>();

10 result.add(totalValue);

11 result.add(finalItems);

12 return result;

13 }

14 }

15

```
10 # Write your code here
11
12 # Print
13 # Print the first element of the list
14 # Print the last element of the list
15 # Print the second element of the list
16 # Print the third element of the list
17 # Print the fourth element of the list
18 # Print the fifth element of the list
19 # Print the sixth element of the list
20 # Print the seventh element of the list
21 # Print the eighth element of the list
22 # Print the ninth element of the list
23 # Print the tenth element of the list
```

Run or submit code when you're ready.