weights = {10, 8, 4, 3, 2, 1} capacity: 15, size = 3

expected = [{10, 4, 1}, {10, 3, 2}]

|  |  |  |  |
| --- | --- | --- | --- |
| in | weights | capacity | size |
|  | 10, 8, 4, 3, 2, 1 | 15 | 3 |
| 10 | 8, 4, 3, 2, 1 | 5 | 2 |
| 8 reject |  |  |  |
| 10, 4 | 3, 2, 1 | 1 | 1 |
| 3, 2 reject |  |  |  |
| 10, 4, 1 | NEW SET |  |  |
| Back to: |  |  |  |
| 10 | 8, 4, 3, 2, 1 | 5 | 2 |
| 8, 4 reject |  |  |  |
| 10, 3 | 2, 1 | 2 | 1 |
| 10, 3, 2 | NEW SET |  |  |

weights = {10, 8, 7, 3, 1} capacity: 20, size = 3

expected = [{10, 7, 3}]

|  |  |  |  |
| --- | --- | --- | --- |
| in | weights | capacity | size |
|  | 10, 8, 7, 3, 1 | 20 | 3 |
| 10 | 8, 7, 3, 1 | 10 | 2 |
| 10, 8 | 7, 3, 1 | 2 | 1 |
| None fits,  Go back to: |  |  |  |
| 10 | 8, 7, 3, 1 | 10 | 2 |
| 8 reject |  |  |  |
| 10, 7 | 3, 1 | 3 | 1 |
| 10, 7, 3 | NEW SET |  |  |

weights = [4, 3, 2, 1], capacity = 5, size = 2

expected = [[4, 1], [3, 2]]

for weight in weights:  
 weights\_after\_weight = weights[weights.index(weight) + 1:].copy()  
 result = choices\_by\_size(capacity - weight, size - 1, weights\_after\_weight)  
 if result is not None:  
 for value, choice in result:  
 if capacity >= value + weight >= max\_value:  
 max\_value = value + weight  
 max\_choice = [weight] + choice  
 ret\_list.append([max\_value, max\_choice])

weight = 4, weights\_after\_weight = [3, 2, 1]

result = choices\_by\_size(capacity=1, size=1, weights=[3, 2, 1])

weight = 3, after = [2, 1]

result = choices\_by\_size(capacity = -2…), so next weight

weight = 2, after = [1]

result = choices\_by\_size(capacity = -1…), so next weight

weight = 1, after = []

result = choices\_by\_size(capacity =0, size=0, weights=[])

result = [[0, []]]

value = 0, choice = []

max\_value = 1, max\_choice = [1]

result = [[1, [1]]]

value = 1, choice = [1]

max\_value = 5, max\_choice = [4, 1] ret\_list = [[5, [4,1]]]

for weight in weights:  
 weights\_after\_weight = weights[weights.index(weight) + 1:].copy()  
 result = choices\_by\_size(capacity - weight, size - 1, weights\_after\_weight)  
 if result is not None:  
 for value, choice in result:  
 if capacity >= value + weight >= max\_value:  
 max\_value = value + weight  
 max\_choice = [weight] + choice  
 ret\_list.append([max\_value, max\_choice])

ret\_list = [[5, [4,1]]]

weight = 3, after = [2, 1]

result = choices\_by\_size(capacity=2, size=1, weights=[2, 1])

weight = 2, after [1]

result = choices\_by\_size(capacity=0, size=0, weights=[1])

result = [[0, []]]

value = 0, choice = []

max\_value = 2, max\_choice = [2] ret\_list = [[2, [2]]]

weight = 1, after = []

result = choices\_by\_size(capacity=1, size=0, weights=[]), so None