

```
1 # Python code for integer knapsack problem, with numpy
2
3 import numpy
4
5 hugeNumber = float("inf")
6 unused = -1000 # Dummy value to make code more readable
7
8 items = 3
9 capacity = 10
10
11 itemWeight = numpy.array([unused, 4, 3, 5]) # "unused" is for element 0,
12 itemValue = numpy.array([unused, 11, 7, 12]) # which we don't use
13
14 f = numpy.zeros([items + 2, capacity + 1])
15 x = numpy.zeros([items + 1, capacity + 1], dtype=int)
16
17 for t in range(items, 0, -1) : # Loop items, items-1, ... , 1
18
19     for i in range(capacity+1) : # Loop over all possible amounts
20         # of space left (0 to capacity)
21         maxCanFit = int(i/itemWeight[t]) # Figure out how many will fit (int
22         rounds down)
23
24         value = -hugeNumber # Because we're maximizing
25
26         for d in range(maxCanFit + 1) : # Loop d from 0 to maxCanFit
27
28             j = i - d*itemWeight[t]
29             moveValue = d*itemValue[t] + f[t+1,j]
30
31             if moveValue > value : # > here because we're maximizing
32                 value = moveValue
33                 bestMove = d
34
35         # End of d loop
36
37         f[t,i] = value
38         x[t,i] = bestMove
39
40     # End of i loop
41 # End of t loop
42
43 print("Optimal value is " + str(f[1,capacity]))
44
45 print("Item counts:")
46 i = capacity
47 for t in range(1,items+1) :
48     print(str(x[t,i]) + " of type " + str(t))
49     i = i - x[t,i]*itemWeight[t]
50 print("Unused space is " + str(i))
51
```