'''

Problem:

0 1 package problem.

Giben n objects with weights {w1, w2, ..., wn} and values {v1, v2, ..., vn},

Suppose we have a package which can hold the weight N at most,

Which objects should be put into the package, so as to obtain the most total value?

Ways:

dynamic program

construct the object-volume table from bottom to top, from left to right

formulation: f[i,j] = max{f[i-1,j], f[i+1,j-weights[i]]+values[i]}

Requires:

Ref:

https://blog.csdn.net/mu399/article/details/7722810

'''

import numpy as np

def get\_01\_package(weight, value, bag\_size):

num\_obj = len(weight)

arr\_val = np.zeros((num\_obj, bag\_size))

# loop each size

for jcol in range(bag\_size):

# loop each object

for irow in range(num\_obj-1,-1, -1):

if weight[irow] > jcol+1:

# package irow can not hold item

if irow == num\_obj-1:

arr\_val[irow, jcol] = 0

else:

arr\_val[irow, jcol] = arr\_val[irow+1, jcol]

else:

# package irow can hold item

if irow == num\_obj - 1:

arr\_val[irow, jcol] = value[irow]

else:

value\_in\_item = value[irow] + arr\_val[irow+1, jcol-weight[irow]]

arr\_val[irow, jcol] = value\_in\_item if value\_in\_item > arr\_val[irow+1, jcol] else arr\_val[irow+1, jcol]

return arr\_val

bag\_size = 12

weight = [2, 2, 6, 5, 4]

value = [6, 3, 5, 4, 6]

print(get\_01\_package(weight, value, bag\_size))