import numpy as np

def zero\_one\_package(cost, worth, bagsize, f):

for isize in range(bagsize, cost-1, -1):

worth\_in\_item = f[isize - cost] + worth

if worth\_in\_item > f[isize]:

f[isize] = worth\_in\_item

def complete\_package(cost, worth, bagsize, f):

for isize in range(cost, bagsize+1):

worth\_in\_item = f[isize - cost] + worth

if worth\_in\_item > f[isize]:

f[isize] = worth\_in\_item

def complete\_package\_2d(worth, cost\_u, cost\_v, bagsize\_u, bagsize\_v, f):

for usize in range(cost\_u, bagsize\_u+1):

for vsize in range(cost\_v, bagsize\_v+1):

worth\_in\_item = f[usize-cost\_u, vsize-cost\_v] + worth

if worth\_in\_item > f[usize, vsize]:

f[usize, vsize] = worth\_in\_item

def multiple\_package(cost, worth, amount, bagsize, f):

if cost \* amount >= bagsize:

complete\_package(cost, worth, bagsize, f)

else:

k = 1

while k < amount:

zero\_one\_package(cost\*k, worth\*k, bagsize, f)

amount -= k

k \*= 2

zero\_one\_package(cost\*amount, worth\*amount, bagsize, f)

# notice, first loop the bagsize, then loop the object in group

def group\_package(worth, cost, bagsize, f):

num\_obj = len(worth)

for isize in range(bagsize, -1, -1):

for iobj in range(num\_obj):

if isize >= cost[iobj]:

value\_in\_item = f[isize-cost[iobj]] + worth[iobj]

if value\_in\_item > f[isize]:

f[isize] = value\_in\_item

bagsize = 10

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

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

obj\_num = len(cost)

# print('zero one package')

# f = [0] \* (bagsize + 1)

# for iobj in range(obj\_num):

# zero\_one\_package(cost[iobj], worth[iobj], bagsize, f)

# print(f[1:])

# print('complete package')

# f = [0] \* (bagsize + 1)

# for iobj in range(obj\_num):

# complete\_package(cost[iobj], worth[iobj], bagsize, f)

# print(f[1:])

# amount = [3, 2, 3, 1, 4]

# print('multiple package')

# f = [0] \* (bagsize + 1)

# for iobj in range(obj\_num):

# multiple\_package(cost[iobj], worth[iobj], amount[iobj], bagsize, f)

# print(f[1:])

# worth = [6, 3, 5, 4, 6]

# obj\_num = len(cost)

# bagsize\_u = 10

# cost\_u = [2, 2, 6, 5, 4]

# bagsize\_v = 8

# cost\_v = [3, 1, 4, 4, 2]

# print('multiple cost')

# f = np.zeros((bagsize\_u+1, bagsize\_v+1))

# for iobj in range(obj\_num):

# complete\_package\_2d(worth[iobj], cost\_u[iobj], cost\_v[iobj], bagsize\_u, bagsize\_v, f)

# print(f[1:, 1:])

worth1 = [3, 4, 6, 5, 2]

cost1 = [1, 3, 2, 4, 4]

worth2 = [4, 3, 5, 2]

cost2 = [3, 2, 4, 4]

worth3 = [6, 2, 5, 4, 3]

cost3 = [4, 4, 2, 1, 3]

bagsize = 10

print('group object')

f = [0] \* (bagsize + 1)

worth\_cost = ((worth1, cost1), (worth2, cost2), (worth3, cost3))

for igroup in range(len(worth\_cost)):

data\_temp = worth\_cost[igroup]

worth = data\_temp[0]

cost = data\_temp[1]

group\_package(worth, cost, bagsize, f)

print(f[1:])