'''

Problem:

zero one package problem

Requires:

the space complexity should be O(N)

Ways:

(1)for the loop about bagsize, from large to small

(2)f[i] means the maximum value when select from the first i objects

f[0] means the value for no object

(3)the initialization corresponding to whether require the bag to be full

'''

import copy

def sum\_cost(cost):

cost\_sum = copy.deepcopy(cost)

for iobj in range(len(cost) - 2, -1, -1):

cost\_sum[iobj] += cost\_sum[iobj + 1]

return cost\_sum

# version 4

# restrict the bag must be full

def zero\_one\_package(cost, value, bagsize):

obj\_num = len(value)

f = [-1000000] \* (bagsize + 1)

f[0] = 0

cost\_sum = sum\_cost(cost)

for iobj in range(obj\_num):

# ensure the volume of bag can hold the object

bound = max(bagsize-cost\_sum[iobj], cost[iobj])

for jsize in range(bagsize, bound - 1, -1):

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

if f\_in\_item > f[jsize]:

f[jsize] = f\_in\_item

return f[1:]

# # version 3

# # optimize the low bound of jsize further

# def zero\_one\_package(cost, value, bagsize):

# obj\_num = len(value)

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

# cost\_sum = sum\_cost(cost)

# for iobj in range(obj\_num):

# # ensure the volume of bag can hold the object

# bound = max(bagsize-cost\_sum[iobj], cost[iobj])

# for jsize in range(bagsize, bound - 1, -1):

# f\_in\_item = f[jsize-cost[iobj]] + value[iobj]

# if f\_in\_item > f[jsize]:

# f[jsize] = f\_in\_item

# return f[1:]

# # version 2

# # optimize the low bound of jsize

# def zero\_one\_package(cost, value, bagsize):

# obj\_num = len(value)

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

# for iobj in range(obj\_num):

# # ensure the volume of bag can hold the object

# for jsize in range(bagsize, cost[iobj]-1, -1):

# f\_in\_item = f[jsize-cost[iobj]] + value[iobj]

# if f\_in\_item > f[jsize]:

# f[jsize] = f\_in\_item

# return f[1:]

# # version 1

# def zero\_one\_package(cost, value, bagsize):

# obj\_num = len(value)

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

# for iobj in range(obj\_num):

# for jsize in range(bagsize, -1, -1):

# # check the volume of bag can hold the object

# if jsize >= cost[iobj]:

# f\_in\_item = f[jsize-cost[iobj]] + value[iobj]

# if f\_in\_item > f[jsize]:

# f[jsize] = f\_in\_item

# return f[1:]

bagsize = 12

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

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

result = zero\_one\_package(cost, value, bagsize)

print(result, len(result))