1.

from functools import reduce

def knapsack1(fruits, limit):

def nextVI(i, values, items):

return reduce(

(lambda vis, vi: (vis[0] + [vi[0]], vis[1] + [vi[1]])),

[(values[w - fruits[i][1]] + fruits[i][2], i)

if w >= fruits[i][1] and w < limit + 1 and

values[w - fruits[i][1]] + fruits[i][2] > values[w]

else (values[w], items[w]) for w in range(len(values))],

([], [])

)

def iterate(i):

if i == 0:

return nextVI(i, [0] \* (limit + 1), [0] \* (limit + 1))

else:

values, items = iterate(i - 1)

return nextVI(i, values, items)

def solution(i, items, minWeight):

return (([fruits[items[i]]] +

solution(i - fruits[items[i]][1], items, minWeight))

if i >= minWeight else [])

return solution(limit, iterate(len(fruits) - 1)[1], min([f[1] for f in fruits]))

print(knapsack1([('釋迦', 5, 800), ('西瓜', 3, 200),

('玉荷包', 2, 600), ('蘋果', 2, 700),

('黑金剛(蓮霧)', 3, 400),

('番茄', 1, 100)], 5))

def knapsack2(W, wt, val):

n = len(val)

table = [[0 for x in range(W + 1)] for x in range(n + 1)]

for r in range(n + 1):

for c in range(W + 1):

if r == 0 or c == 0:

table[r][c] = 0

elif wt[r-1] <= c:

table[r][c] = max(val[r-1] + table[r-1][c-wt[r-1]], table[r-1][c])

else:

table[r][c] = table[r-1][c]

return table[n][W]

value = [800, 200, 600, 700, 400, 100]

weight = [5, 3, 2, 2, 3, 1]

bag\_weight = 5

print('商品價值:', knapsack2(bag\_weight, weight, value))