Two Dimension Knapsack

二维背包

问题：

你面前摆放着n个珠宝（共n种，每种 1 个），已知珠宝s_i的价值是v_i，重量 1 是$w1_i$，重量 2 是$w2_i$。给你一个背包，你可以自由挑选珠宝装到背包中，但背包可以装载的最大重量 1 为$W1$，最大重量 2 为$W2$。求背包能够装载珠宝的最大价值v。

该问题与 01 背包的区别就是，重量属性变成了 2 维属性，背包中所有珠宝的总重量 1 不能超过$W1$，总重量 2 不能超过$W2$。

解法：

设$f(i,j,k)$为背包中放入前i件物品，重量 1 不大于j，重量 2 不大于k的最大价值，其中$i \in [1,n]$，$j \in [0,W1]$，$k \in [0,W2]$。有如下状态转移方程：

$$f(i,j,k) = \begin{cases} 0 & \text{（初始化）} i = 0 \\ f(i-1,j,k) & i,j,k > 0 \\ max(f(i-1,j,k), f(i-1,j-w1_i,k-w2_i)+v_i) & i,j,k > 0\text{，} j \geq w1_i\text{，} k \geq w2_i \end{cases}$$

(1) 用数组中的下标 0 来存储初始的固定值，背包中没有放入任何珠宝时，$f(0,j)=0$；
(2) 对于第i件珠宝s_i，背包的剩余重量 1（还能装载的重量）为$W1$，剩余重量 2 为$W2$，若$W1 \geq k \times w1_i$，$W2 \geq k \times w2_i$，那么可以装进 1 个珠宝s_i，背包的价值增大v_i，剩余重量 1 减小$w1_i$，剩余重量 2 减小$w2_i$即$f(i,j,k) = f(i-1,j-w1_i,k-w2_i)+v_i$；若不装入背包，则一切维持不变，即$f(i,j,k)=f(i-1,j,k)$。选择这两种情形中的最大值；

$f(n,W1,W2)$即为n个珠宝中重量 1 不超$W1$，重量 2 不超过$W2$的最大价值。该算法的时间复杂度是$O(n \times W1 \times W2)$。