Group Knapsack

分组背包

问题：

你面前摆放着n个珠宝（共n种，每种 1 个），这些珠宝被分成m个组（显然$n \geq m$）。已知珠宝s_i的价值是v_i，重量是w_i。给你一个背包，你可以挑选珠宝装到背包中，但背包可以装载的最大重量为t，并且同一个组的珠宝只能选择 1 个。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，对珠宝进行了分组，并且一个组内的珠宝互斥。

解法：

设$f(i,j)$为背包中放入前i组物品，重量不大于j的最大价值，其中$i \in [1,m]$，$j \in [0,t]$。第i组中有$group_i$个珠宝，其中某珠宝k的价值是v_k，重量是w_k。则有如下状态转移方程：

$$f(i,j) = \begin{cases} 0 & \text{（初始化）} i = 0 \\ f(i-1,j) & i,j > 0 \\ max(f(i-1,j), f(i-1,j-w_k)+v_k) & i,j > 0\text{，} k \in [1, group_i]\text{，} j \geq w_k \end{cases}$$

(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组珠宝，背包的剩余重量（还能装载的重量）为W，在第i组珠宝中选择某个珠宝k，若$W \geq w_k$，那么可以装进珠宝k，背包的价值增大v_k，剩余重量减小w_k，即$f(i,j) = f(i-1,j-w_k) + v_k$；若不装入背包，则一切维持不变，即$f(i,j) = f(i-1,j)$。选择这两种情形中的最大值；

$f(m,t)$即为m组珠宝中重量不超过t的最大价值。该算法的时间复杂度是$O(n \times t)$。