### 穷举法解决MKP问题的思路分析

**代码思路分析：**

首先要了解了测试数据的结构，

测试数据结构：

6 10 3800 //n物品个数 m约束条件个数（背包个数） best最优解决方案值

100 600 1200 2400 500 2000 //价值数组

8 12 13 64 22 41 //m行n列维度（或者说质量）的二维数组

8 12 13 75 22 41

3 6 4 18 6 4

5 10 8 32 6 12

5 13 8 42 6 20

5 13 8 48 6 20

0 0 0 0 8 0

3 0 4 0 8 0

3 2 4 0 8 4

3 2 4 8 8 4

80 96 20 36 44 48 10 18 22 24 //每一行（每一维度）的背包上限（约束条件）

在确定了解了数据的结构后，

先把所有物品的选取情况都遍历出来

在这个MKP问题里那就是物品的放和不放的问题，这个可以用0和1来表达，0代表不选取该物品，1代表选取该物品，那一组数就是0和1的任意位数的排列结果。

例如：当有6个物品，n=6时，物品的选取情况可以用以下来表示

000000 000001 000010 000011 000100 000101 000110 ....直到 111111

n个物品那就有2^n种排列结果。

然后我们先把第一个排列结果都依次存储到一个一维数组里，然后**判断**该物品选取方案（即当前存储01排列结果的一维数组）下所有维度（即每一组属性）放入背包的物品的质量总和是否超出背包上限，

1. 如果没有超出，那就将该物品选取方案下的价值总和与最大值相比较，记录最大值者的路径（下标）
2. 如果这m个维度有一个超出那就立刻退出，测试下一个排列结果（物品选取方案）

以此类推，直到循环测试到2^n个排列结果（物品选取方案）后才结束

最后输出最优解决方案的下标以及其价值总和。

**时间复杂度分析：**

因为n个物品那就有2^n种排列结果，所以穷举法的时间复杂度就是O(2^n)，又因为是多背包问题，有m个属性，所以整个代码的时间复杂度为O(m\*2^n)。