多维背包问题算法分析

**问题分析**

多维背包问题就是有m组背包，每个背包有各自的约束条件，而每个约束只是约束一组背包的物品，例如第一个约束是约束第一组的，不是约束第二组的，第二组有第二个约束条件来约束，每组有n个物品，每个物品拥有一个价值，在保证m组背包n个物品都能在约束条件下选取其中每组背包里面选取相同次序的几个物品来获得最大的价值，例如第一组的第一个、第五个和第六个物品选取了，则下面的组中这一、五、六三个物品也得选，就是在m组中找到同时符合各自约束条件的相同次序的物品来获得一个最大价值。

**算法分析**

解决多维背包问题我采用了回溯法这个算法，回溯法需要构造解的子集树，对于每一个物品只有选与不选2个决策，总共有m组，每组有n个物品，可以顺序依次考虑每组的第1……n个物品，若有一组的第N个物品不满足约束，装不进背包，则当下不考虑第N个物品，继续向下寻找，最后回溯来遍历所有的选与不选的所有可能选出最大的价值方案。

 基本思想就是遍历树，以枚举所有情况，最后进行判断，如果重量不超过背包容量，背包价值最大的话，则该选取的物品方案就是最终方案。

**算法复杂度分析**

当每组有n个数，有m组时。

算法时间复杂度为O（m\*2^n）

因为物品只有选与不选2个决策，而总共有m个组，每组有n个物品，

所以时间复杂度 O（m\*2^n）。

空间复杂度为 O（n^2）

存储所有输入的物品信息需要二维数组，所以空间复杂度为O(n^2)。