多维背包问题其实是一个经过拓展的01背包问题，把原来的重量这一个约束增加到10个约束，并且给出10个约束物品属性的值，并且找到最大价值的物品。以下就是网站中的第一组数据并整理了如下：

**物品、约束、目标价值：6 10 3800**

**v价值: 100 600 1200 2400 500 2000**

**属性1: 8 12 13 64 22 41**

**属性2: 8 12 13 75 22 41**

**属性3: 3 6 4 18 6 4**

**属性4: 5 10 8 32 6 12**

**属性5: 5 13 8 42 6 20**

**属性6: 5 13 8 48 6 20**

**属性7: 0 0 0 0 8 0**

**属性8: 3 0 4 0 8 0**

**属性9: 3 2 4 0 8 4**

**属性10: 3 2 4 8 8 4**

**m约束: 80 96 20 36 44 48 10 18 22 24**

然后发现可以用三个数组存放数据，其中两个一维数组存放物品的价值以及约束值，剩下的十个约束可以用二维数组来存放，一开始我考虑的是用n个一维数组存放，这样可以很快的遍历，但是它所需的空间量就会大大的增加并且使整体的代码很不美观。

利用回溯法就是要求出一个解向量即是对物品放与不放的方案，定义一个变量r来控制判断属性时候符合约束的层数，当r<n时，从根节点出发在二叉树上一直搜索到叶子节点得到一种方案；当r=n时又回溯退回开始另一种方案，直到把整个二叉树遍历完，最后比较各个方案的价值，最大的就是所要求的目标最大价值即3800

因为物品只有选和不选，总共有n个物品和m个属性约束，所以时间复杂度为O（2n\*m）

最大就需要一个二维数组存放属性，所以空间复杂度为O（n\*m）