**算法分析**

**1．实验目的**

使用背包算法，实现装入物品价值最大化

**2．实验内容**

给定一个txt文件，文件包含以下内容：1、n物品数量 2、m背包限制向量的维数 3、预期最大价值 4、每个物品对每个约束的消耗 5、每个物品的价值 6、背包的限制向量，m维

首先读取文件，获取以上信息，根据已有信息，要求装入物品到背包中，切不超出背包限制，价值达到预期价值。

1. **实验原理**

已知：物品数量：n;限制维数：m；预期价值：V；每个物品i对每个维度限制j的消耗：rij；

每个物品的价值vi；背包的限制向量uj;第i个物品是否放入xi；

1）Sum（ i=0->i=n）[ vi\*xi ]=V；

2）Sum（ i=0->i=n）[(j=0->j=m)(rij<ui)];

3) xi=0or1;

实验的核心就是求出xi向量的值，我们通过回溯法来求解。

1. **算法分析**

回溯法是通过建立二叉树，遍历其值最大的节点来求解问题的，设问题的序列为xi，i<n，先建立二叉树的树头，然后依次从i=0到i=n建立二叉树的子节点，设定左子树xi=0，右子树xi=1，建立树的条件为当前消耗是否符合父节点的限制。当建立完二叉树后，遍历其叶子节点，来寻找其最大值点。

1. **算法复杂度**

回溯法分析分为两个部分：

（1）建立二叉树

建立二叉树左子树不需要计算上界，右子树需要计算上界，一维情况下上界计算的时间复杂度为0（n），多维情况下为O（nm），最坏的情况需要计算O（2^n)次右子树的上界限，所以综合来说时间复杂度为O（nm2^n)。空间复杂度为最坏情况下子树个数0（n^2).

（2）回溯

回溯时的时间复杂度为0（n），空间复杂度为O（n^2)。

总体来说，当n较大时，并且情况不理想时，回溯法的时间复杂度还是挺高的，我们可以通过对遍历节点加以约束来优化。