算法设计与分析文档

张梦华 201706401229 计172班

这次的作业查阅大量的资料，并找寻相类似的解法，学习有关知识，也问过其他同学。

此算法类似于0-1背包，属于找最优解问题，用回溯法构造解的子集树。只是这里并不是只有一个约束，而是有多个约束，也就是我们需要解决的问题。对于每一个物品i，对于该物品只有选与不选2个决策，总共有n个物品，可以顺序依次考虑每个物品，这样就形成了一棵解空间树： 基本思想就是遍历这棵树，以枚举所有情况，最后进行判断，如果重量不超过背包容量，且价值最大的话，该方案就是最后的答案。在搜索状态空间树时，只要左子节点是可一个可行结点，搜索就进入其左子树。对于右子树时，先计算上界函数，以判断是否将其减去（剪枝）。

在递归函数Backtrack中，当i>n时，算法搜索至叶子结点，得到一个新的物品装包方案。此时算法适时更新当前的最优价值。当i<n时，当前扩展结点位于排列树的第（i-1）层，此时算法选择下一个要安排的物品，以深度优先方式递归的对相应的子树进行搜索，对不满足上界约束的结点，则剪去相应的子树。

因为物品只有选与不选2个决策，而总共有n个物品，所以时间复杂度为IMG_256。因为递归栈最多达到n层，在每一层下面要判断该物品是否满足条件，是否能放进去则又要循环m层，则函数的复杂性为O(n×m)。在输入的测试数据中，有n个物品和m个约束，x为最大价值。接下来输入每个物品的价值，然后依次输入那些物品的约束值，最后一行则是每个物品的约束值。根据回溯算法获得最优算法，取得最优价值。