Homework 4

和泳毅 PB19010450

1.假定你希望兑换外汇，你意识到与其直接兑换，不如进行多种外币的一系列兑换，最后兑换到你想要的那种外币，可能会获得更大收益。假定你可以交易 n 种不同的货币，编号为 $1,2,\ldots,n$，兑换从 1 号货币开始，最终兑换为n号货币。对每两种货币 i 和 j，给定汇率 r_{ij}。意味着你如果有 d 个单位的货币i，可以兑换 dr_{ij}个单位的货币j 。进行一系列的交易需要支付一定的佣金，金额取决于交易的次数。令c_k表示k次交易需要支付的佣金。证明：如果对所有$k=1,2,\ldots,n,c_k=0$，那么寻找最优兑换序列的问题具有最优子结构。然后请证明：如果佣金 c_k 为任意值，那么问题不一定具有最优子结构。

答:

- 当佣金为0时：

  假设货币k是最优兑换序列其中的一个，基于k可以将最优序列分为$\{i\ldots k\}$和$\{k\ldots j\}$，分别是将货币i兑换为k以及将货币k兑换为j的最优序列。分别记上述两个序列为p_{ik}和p_{kj}，假设此时p_{ik}不是i货币兑换k货币的最优序列，不妨设p'_{ik}为最优序列。那么序列$p'_{ik}p_{kj}$得到的货币j的数量一定比序列$p_{ik}p_{kj}$要多，产生矛盾，故问题具有最优子结构。

- 当佣金为任意值时：

  此时问题不一定具有最优子结构，存在如下反例：

  设有五种货币$\{1,2,3,4,5\}$，其中$r_{12}=r_{23}=r_{34}=r_{45}=2$，$r_{13}=r_{35}=6$，其余$r_{ij}=1$。对于$c_k$，$c_2=1$，其余$c_k=5$。
  若初始货币1为1个单位，货币1兑换货币5的最优解是：$1\to3\to5$，总共获得35个单位的货币5。但是货币1兑换货币3的最优解为$1\to2\to3$，货币3兑换货币5的最优解为$3\to4\to5$，不具有最优子结构。

2.设计一个高效的算法，对实数线上给定的一个点集$\{x_1,x_2,\ldots x_n\}$,求一个**单位长度闭区间**的集合，包含所有给定的点，并要求此**集合最小**。证明你的算法是正确的。

答:

算法步骤：

1. 对点集$\{x_1,x_2,\ldots x_n\}$做快排，使得点集从小到大有序。
2. 从最小的点开始，以该点为起点构造一个单位长度的闭区间。
3. 从点集中删去被包括在区间内的点，返回第2步，直到没有剩余点为止。

证明正确性：

首先假设一个最优解，其由若干闭区间构成，那么由于其为最优解，最小的点必然包含在它最左的区间内，假设最小点是x_0，这个区间为 $[a,b]$，如果这个点并不是区间的左端点，由于x_0本身是最小点，那么 $[a,x0)$本身应当不含任何点，所以修改$[a,b]$为$[x_0,x_0+1]$的话，这个区间集也依然是最优解，也就是说算法的第 2 步所构造的解依然属于最优解之一。接下来分析第 3 步递归，在这一修改过的最优解的基础上，从点集中取出包含在 $[x_0,x_0+1]$ 的点，同时从最优解中去掉 $[x_0,x_0+1]$，则现在的区间集合依然为修改后的点集的最优解，证明算法的第 3 步递归不会破坏构造的解属于最优解之一的特性，所以该算法最终找到的解是最优解。

3.一位公司主席正在向 Stewart 教授咨询公司聚会方案。公司的内部结构关系是层次化的，即员工按主管-下属关系构成一棵树，根结点为公司主席。人事部按“宴会交际能力”为每个员工打分，分值为实数。为了使所有参加聚会的员工都感到愉快，主席不希望**员工及其直接主管**同时出席。 公司主席向 Stewart 教授提供公司结构树，采用**左孩子右兄弟**表示法（参见课本 10.4 节）描述。每个节点除了保存指针外，还保存员工的名字和宴会交际评分。设计算法，求宴会交际**评分之和最大**的宾客名单。分析算法复杂度。

答：

每次对于一个结点（主管）以及它的子树（下属），维护两个属性：

- 一个是该主管出席时，以该主管结点为根的子树所能具有的最大的交际评分和，记作 $v.a$
- 一个是该主管不出席时该树的最大交际评分和，记作 $v.b$

对于叶结点， $v.a$ 即其交际评分， $v.b$ 取0。而对于非叶子结点，它的两个属性可作如下计算：$v.a$为其所有孩子的b属性之和， $v.b$ 为其所有孩子的 a、b 属性最大值之和。直观上看也是合理的，主管出席时的最大交际评分和就是其每一个下属不出席时最大交际评分之和，主管不出席时的最大交际评分和就是其每一个下属出席时最大交际评分之和。

在计算这两个属性的过程中，为了能够输出最佳方案对应的宾客名单，再对每一个结点额外维护一个flag属性，该属性记录的是，当它的主管未被选中时，以它的主管为根结点的子树取到最大交际评分时该结点是否被选中，如果被选中则记录 True，反之记录 False，这个属性可以在计算 a,b属性的同时进行维护，每个结点维护一次。

最后，公司主席根结点的a、b 属性最大者即此次宴会最佳名单对应的交际评分和。

在输出最优解宾客名单时，自顶向下递归：

1. 对根结点分析，如果其 a 属性大于b 属性，那么该主管加入宾客名单，进入步骤2。反之进入步骤3。
2. 所有该结点的下属结点不参加宴会。分别将这些下属结点作为新的主管结点，将其新的下属节点中flag=True的加入宾客名单，对这些结点进行步骤2。对其下属节点中flag=False的进行步骤3。

3. 所有该结点的下属结点可以参加宴会。分别将这些下属结点作为新的主管结点，将其中flag=True的加入宾客名单，对这些结点进行步骤2。对其中flag=False的进行步骤3。

计算属性本质上是做了一轮树的遍历，而自顶向下递归本质上也是做了一轮遍历，两轮遍历中对于每一个结点的操作都是常数时间，所以设人员总数为n，实际计算复杂度是$\Theta(n)$ 。

4.考虑用**最少的硬币**找 n 美分零钱的问题。假定每种硬币的面额都是整数。设计**贪心算法**求解找零问题，假定有 25 美分、10 美分、5 美分和 1 美分四种面额的硬币。证明你的算法能找到最优解。

答:

与背包问题思想类似。采用贪心算法，总是优先尝试用最大面值的硬币来找，找到可能的最大的硬币后，再对剩下需要找的钱进行递归。

最优解证明:

分析递归的过程。每次算法找到当前需要找零数下能使用的最大的硬币，把它从待找金额中减去后再处理剩下需要找的钱。本质上是把一个问题分割成了两个子问题，其中子问题 A 是一个待找金额为该硬币面值的找零问题，子问题 B 是一个待找金额为剩下钱数的找零问题。而分割后的子问题 AB 中，显然 A 在算法里直接得到了最优解硬币数，而假如 B 也找到了最优解，下面证明这两个子问题的最优解合起来是原问题的最优解。

观察两个子问题，如果原问题存在一个最优解，将其内部硬币按面额从大到小排序，其中面额最大的硬币$value_1$ 不超出子问题 A 的值$value_A$。

- 如果$value_1 = value_A$，那么该算法对于问题 AB的划分包含了最优解。
- 如果$value_1 < value_A$，则该最优解集合中的所有硬币面额都小于$value_A$。在最优解集合中，面额5最多只有1个，否则可用面额10来代替。同理面额1最多有4个，面额10最多两个，面额25数量不限。从而我们发现，只使用面额1的最优解，最多能表示4。只使用面额1、5的最优解，最多能表示9。只使用面额1、5、10的最优解，最多能表示24。所以在本题的币值中，最优解不使用面额大于等于$value_A$的硬币就无法表示大于等于$value_A$的面额，即$value_A$大于最优解面额之和，这是矛盾的，故$value_1$不小于$value_A$。

所以，该贪心算法只要 B 子问题找得到最优解，那么算法本身必然找到的是最优解，由其递归性可知， B 子问题的分割必然会在某一步变成单硬币的可解问题，所以该贪心算法总是能找到最优解。