

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 $1, 2, \dots, n$ ，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 $k = 1, 2, \dots, n$, $c_k = 0$ ，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。

对于从货币 i 兑换为货币 j ，假设第一步先将货币 i 兑换为货币 k ，则接下来只需求将货币 k 兑换为货币 j 可获得的最大收益，在从货币 k 兑换为货币 j 的过程中，可选的货币集合不包括 i 、 k 、 j 。假设将货币 k 兑换为货币 j 获得的收益不是最大的，则存在比此更优的方案，使得收益更大。

如果佣金 c_k 为任意值，那么问题不一定具有最优子结构：考虑 $r_{12} = r_{13} = r_{24} = r_{34} = 2$, $r_{32} = 1.5$ ，对于其他的 r_{ij} , $r_{ij} = 1$, $c_1 = c_2 = 0$, $c_3 = 100$ ，此问题的最优方案为 $1 \rightarrow 2 \rightarrow 4$ ，而对于 $2 \rightarrow 4$ ，有更优的方案： $2 \rightarrow 3 \rightarrow 4$ ，但由于 $c_3 = 100$ ，故方案 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ 不是最优的。由上可以看出，虽然子结构 $k \rightarrow j$ 的某一方案为最优的，但会受到支付佣金的影响，导致整体的收益可能会有所下降。

2. 设计一个高效的算法，对实数线上给定的一个点集 $\{x_1, x_2, \dots, x_n\}$ ，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。

1) 将所求的区间 S 初始化为空集，用快速排序将点集按升序排序

2) 每次选出目前点集中的最小值，取以该点为左起点的单位长度闭区间 Q ，将 Q 加入区间 S 中，在点集中除去包含在 Q 中的点。

3) 重复步骤 2) 直到点集为空

每次做出贪心选择后，只剩一个子问题：求出包含剩余点集的最小单位长度闭区间的集合，且该子问题总是存在最优解，即贪心选择是安全的。假设 S 为最优解，则一定包含子问题点集 P 中的最小点 m ，则 S 中包含该最小点的单位闭区间 $[s_i, s_i + 1]$ 中，有 $s_i \leq m$ 。用以该点为左起点的单位长度闭区间 $[m, m + 1]$ 代替最优解 S 中的区间 $[s_i, s_i + 1]$ ，由于 $s_i \leq m$ ，且 m 为当前点集的最小点，即在 $[m, s_i]$ 内不会包含点集中除 m 以外的任何点，故得到的新解一定比原来的解更好。

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。

用数组 $y[k]$ 记录以结点 k 为根的子树在 k 参加聚会时的最大交际评分之和，数组 $n[k]$ 记录以结点 k 为根的子树在 k 不参加聚会时的最大交际评分之和，则 k 有两种选择，

k 参加聚会，则 k 的直接下属都不能参加聚会，故 $y[k] = score[k] + \sum n[k_{ci}]$ ，其中 c_i 是 k 的直接下属

k 不参加聚会，则 k 的直接下属可以参加聚会，也可以不参加，故 $n[k] = \sum \max(y[k_{ci}], n[k_{ci}])$ ，其中 c_i 是 k 的直接下属

对于结点 k ，当 $y[k] > n[k]$ 时， k 参加聚会，否则不参加聚会

采用自底向上的算法，先将最底层员工 i 对应的数组初始化， $y[i] = \text{score}[i]$ ， $n[i] = 0$ ，再依次往上计算，当 $y[k] > n[k]$ 时， $\text{attend}[k] = 1$ ，否则设为 0。除去最底层的员工，其他员工的 $y[i]$ 、 $n[i]$ 都需初始化为 $-\infty$ ，并且在循环的内部还需遍历员工的直接下属，故总的时间复杂度为 $O(n^2)$

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。

1)每次取小于 n 美分的最大面额的硬币 money ，再用 n 减去 money 得到新的 n

2)重复步骤1) 直至 n 为 0

每次取小于 n 美分的最大面额的硬币 money 后，只剩一个子问题需要求解：用最少的硬币找 $n - \text{money}$ 的问题，且该子问题一定有最优解，故贪心选择总是安全的。假设不取小于 n 美分的最大面额的硬币，而以较小面额的硬币代替，由于较大面额的硬币总可以由较小面额硬币组成（如 $25=10+10+5$ ， $10=5+5$ 等），故用较小面额的硬币找零所需的个数必然更多。