算法基础 2023 春 Homework 3

任课老师：陈雪 & 邵帅 due: April 20, 23:59

作业要求：说明思路与符号，清晰简洁的伪代码，必要的时间复杂度分析和必要的正确性分析。可以直接调用基本的数据库和已讨论过的算法/程序（如排序、找中位数、二分查找等）。

问题 1 (30 分). 给定一个序列 $a_1, \cdots, a_n$，设计不同的算法来找到最长的递增子序列 $s_1, \cdots, s_m$ 满足如下要求。

1. 如果有多个长度相同的子序列，输出第一个元素最小的序列；如果有多个序列同时有相同的长度和第一个元素，输出第二个元素最小的；以此类推。
2. 如果有多个长度相同的子序列，输出第一个元素最大的序列；如果有多个序列同时有相同的长度和第一个元素，输出第二个元素最大的；以此类推。

HINT: 首先设计一个算法计算最长递增子序列的长度，再输出对应方案。为了获得全部分数，算法的时间复杂度应为 $O(n \log n)$。

问题 2 (20 分). 给定一个树 T 和覆盖半径 r，其中 T 的根节点为 $root$，每个节点 u 的孩子保存在 $c.child$ 序列中。找出 T 中最小的覆盖集（使得每个点到该集合的距离都 $\leq r$）。

问题 3 (25 分). 给定一个背包其重量上线为 M 和 n 个物品，其中每个物品重量为 w_i，价值为 v_i。请选择一些物品使得其总重量不超过 M 的同时总价值最大。

同时回答：算法的时间是否为多项式时间吗？给出必要的理由。

问题 4 (25 分). 为了将一个文本串 $x[1, \cdots, m]$ 转换为目标串 $y[1, \cdots, n]$，我们可以使用多种变换操作。我们的目标是，给定 x 和 y，求将 x 转换为 y 的一个变换操作序列。我们使用一个数组 z 保存中间结果，假定它足够大，可存下中间结果的所有字符。初始时，z 是空的，结束时，应有 $z[j] = y[j], j = 1, 2, \cdots, n$。我们维护两个下标 i 和 j，分别指向 x 中位置和 z 中位置，变换操作允许改变 z 的内容和这两个下标。初始时，$i = j = 1$。在转换过程中应处理 x 的所有字符，这意味着在变换操作结束时，应有 $i = m + 1$。

我们可以使用如下变换操作：

1. 复制：从 x 复制一个字符到 z，即进行赋值 $z[j++] = x[i++]$
2. 替换：将 x 中的一个字符替换为另一个字符 c，即 $z[j++] = c, i++$
3. 删除：删除 x 中一个字符，即 $i++$

4. 插入：将字符 c 插入中 z，即 $z[j++] = c$

每种变换操作每次执行都有一定的代价 $c_{copy}, c_{replace}, c_{delete}, c_{insert}$，而 x 到 y 的编辑距离是将 x 转换为 y 的最小的变换代价之和。设计动态规划算法，输入为文本串 x, 目标串 y 和每种操作的代价 $cost$，求 $x[1, \cdots, m]$ 到 $y[1, \cdots, n]$ 的编辑距离并打印最优操作序列。