算法导论习题选集

作业 7-1

节选自《算法导论》教材第三版

课程网站：https://algorithm.cuijiacai.com

Problem 1

（比较排序的概率下界） 在这一问题中，我们将证明对于给定的 n 个互异的输入元素，任何确定或随机的比较排序算法，其概率运行时间都有下界 $\Omega(n \log n)$ 。首先来分析一个确定的比较排序算法 A ，其决策树为 T_A 。假设 A 的输入的每一种排列情况都是等可能的。

1. 假设 T_A 的每个叶结点都标有在随机输入情况下到达该结点的概率。证明：恰有 $n!$ 个叶结点标有 $1/n!$ ，其他的叶结点标记为 0 。

2. 定义 $D(T)$ 表示一棵决策树 T 的外部路径长度，即 $D(T)$ 是 T 的所有叶结点深度的和。假设 T 为一棵有 $k > 1$ 个叶结点的决策树，LT 和 RT 分别是 T 的左子树和右子树。证明：$D(T) = D(LT) + D(RT) + k$ 。

3. 定义 $d(k)$ 为所有具有 $k > 1$ 个叶结点的决策树 T 的最小 $D(T)$ 值。证明：$d(k) = \min\limits_{1 \le i \le k-1} \{d(i) + d(k-i) + k\}$ 。（提示：考虑一棵有 k 个叶结点的决策树 T 。设 i 是 LT 中的叶结点数，则 $k - i$ 是 RT 中的叶结点数。）

4. 证明：对于给定的 $k(k > 1)$ 和 $i(1 \le i \le k-1)$ ，函数 $i \log i + (k-i)\log(k-i)$ 在 $i = k/2$ 处取得最小值，并有结论 $d(k) = \Omega(k \log k)$ 。

5. 证明：$D(T_A) = \Omega(n! \log(n!))$ ，并得出在平均情况下，排序 n 个元素的时间代价为 $\Omega(n \log n)$ 这一结论。

现在来考虑一个随机化的比较排序 B 。通过引入两种结点，我们可以将决策树模型扩展来处理随机化的情况。这两种结点是：普通的比较结点和“随机化”结点。随机化结点刻画了算法 B 中所做的形如 RANDOM$(1, r)$ 的随机选择情况。该类结点有 r 个子结点，在算法执行过程中，每一个子结点等概率地被选择。

6. 证明：对任何随机化的比较排序算法 B ，总存在一个确定的比较排序算法 A ，其期望的比较次数不多于 B 的比较次数。

Problem 1 continued on next page…

(续页)

(续页)

Problem 2

（线性时间原址排序） 假设有一个包含 n 个待排序数据记录的数组，且每条记录的关键字的值为 0 或 1 。对这样一组记录进行排序的算法可能具备如下三种特性中的一部分：

- 算法的时间代价是 $O(n)$ 。
- 算法是稳定的。
- 算法是原址排序，除了输入数组之外，算法只需要固定的额外存储空间。

1. 给出一个满足上述第 1 个条件和第 2 个条件的算法。
2. 给出一个满足上述第 1 个条件和第 3 个条件的算法。
3. 给出一个满足上述第 1 个条件和第 2 个条件的算法。

4. 第 1 问到第 3 问中的算法中的任一个是否可以用于 RADIX-SORT 的第 2 行作为基础排序方法，从而使 RADIX-SORT 在排序有 b 位关键字的 n 条记录时的时间代价是 $O(bn)$ ？如果可以，请解释应如何处理；如果不行，请说明原因。

5. 假设有 n 条记录，其中所有关键字的值都在 1 到 k 的区间内。你应该如何修改计数排序（见第 7 讲 PPT 第 11 页），使得它可以在 $O(n+k)$ 时间内完成对 n 条记录的原址排序。除输入数组外，你可以使用 $O(k)$ 大小的额外存储空间。你给出的算法是稳定的吗？（提示：当 $k=3$ 时，你该如何做？）

Problem 2 continued on next page…

(续页)

Problem 3

（变长数据项的排序）

1. 给定一个整数数组，其中不同的整数所包含的数字的位数（不含前导 0 ）可能不同，但该数组中，所有整数中包含的总数字位数为 n 。设计一个算法，使其可以在 $O(n)$ 时间内对该数组进行排序。

2. 给定一个字符串数组，其中不同的字符串所包含的字符数可能不同，但所有字符串中的总字符个数为 n 。设计一个算法，使其可以在 $O(n)$ 时间内对该数组排序。（注意：此处的顺序是指标准的字典序，例如 $a < ab < b$ 。）

Problem 4

（**水壶**）假设给了你 n 个红色的水壶和 n 个蓝色的水壶。它们的形状和尺寸都各不相同。所有红色水壶中所盛的水都不一样多，蓝色水壶也是如此。而且，对于每一个红色水壶来说，都有一个对应的蓝色水壶，两者盛有一样多的水；反之亦然。

你的任务是找出所有的所盛水量一样多的红色水壶和蓝色水壶，并将它们配成一对。为此，可以执行如下操作：挑出一对水壶，其中一个是红色的，另一个是蓝色的，将红色水壶中倒满水，再将水倒入蓝色的水壶中。通过这一操作，可以判断出这个红色水壶是否比蓝色水壶盛的水更多，或者两者是一样多的。假设这样的比较需要花费一个单位时间。你的目标是找出一个算法，它能够用最少的比较次数来确定所有水壶的配对。注意，你不能直接比较两个红色或者两个蓝色的水壶。

1. 设计一个确定性算法，它能够用 $\Theta(n^2)$ 次比较来完成所有水壶的配对。

2. 证明：解决该问题算法的比较次数下界为 $\Omega(n\log n)$ 。

3. 设计一个随机算法，其期望的比较次数为 $O(n\log n)$ ，并证明这个界是正确的。对你的算法来说，最坏情况下的比较次数是多少？

Problem 4 continued on next page…

(续页)