第三周作业-solution

LECTURER: 杨启哲 LAST MODIFIED: 2023 年 12 月 23 日

1. (教材习题 5.2) 考虑算法 $SlowMinmax$，它是将算法 $Minmax$ 的检验条件 if $high - low = 1$ 修改为 if $high = low$，并对此算法做一些相应改变而得出的。这样，在算法 $SlowMinmax$ 中，当输入数组的大小为 1 时，递归停止。计算由此算法找出数组 $A[1, \ldots, n]$ 中的最大值和最小值所需要的比较次数，这里 n 是 2 的幂。并解释为什么此算法的比较次数大于算法 $Minmax$ 的比较次数。

   Hint: 在这种情形下，初始条件是 $C(1) = 0$

   解答. 注意到在题设条件下，算法 $SlowMinmax$ 的递归式为

$$C(n) = \begin{cases} 0, & n = 1 \\ 2C(\frac{n}{2}) + 2, & n > 1 \end{cases}$$

   则考察 $n = 2^k$ 我们有:

$$\begin{aligned} C(n) &= 2C(\frac{n}{2}) + 2 \\ &= 2(2C(\frac{n}{4}) + 2) + 2 \\ &\vdots \\ &= 2^k C(\frac{n}{2^k}) + 2(1 + 2 + \cdots + 2^{k-1}) \\ &= 2^k C(1) + 2(1 + 2 + \cdots + 2^k) \\ &= 2(2^k - 1) \\ &= 2n - 2 \end{aligned}$$

   其比较次数大于书上 $MINMAX$ 的算法,这是因为在这个算法中,递归深度更深了一层,$MINMAX$ 递归到 $n = 2$ 就停止了，而 $SlowMinmax$ 则递归到 $n = 1$ 才停止。 □

2. (教材习题 5.26) 对于某个整数 $g \geq 3$，用 g 来表示算法 $Select$ 中每组的规模，导出用 g 表示的算法的运行时间。当 $g = 3, 7, 9, 11$ 时，哪个选择可以保证算法在最坏情况下执行的次数依旧是 $\Theta(n)$?

   解答. 用 g 来代替 5 可得算法的递推式为:

$$T(n) = T(\frac{1}{2g+1}n) + T(\frac{3g+1}{4g+2}n) + cn$$

   注意到 $g = 3$ 时 $\frac{1}{2g+1} + \frac{3g+1}{4g+2} = 1$，而 $g > 5$ 时 $\frac{1}{2g+1} + \frac{3g+1}{4g+2} < 1$，并且该值随 g 增加而递减，从而递归展开可知:

   - 当 $g = 3$ 时 $T(n) = \Theta(n \log n)$。
   - 当 $g = 7, 9, 11$ 时 $T(n) = \Theta(n)$。

   □

Remark 0.1

由上述分析可知，$g = 5$ 时算法有最快的运行速度。

3. (教材习题 5.45) 设 $x = a + bi$ 和 $y = c + di$ 是两个复数。只要 4 次乘法就很容易计算乘积

$$xy = (ac - bd) + (ad + bc)i$$

现在请设计一个方法，只用 3 次乘法去计算 xy.

解答. 这个想法是简单的，我们只需要计算如下的乘积：

- $A = (a + b)(c + d)$
- $B = ac$
- $C = bd$

则我们有：

$$xy = (B - C) + (A - B - C)i$$

□

4. (螺母和螺栓) 有个木匠有一堆混合的 N 个螺母和 N 个螺栓。他希望能找到相应的螺母和螺栓配对。每个螺母只能配对一个螺栓，每个螺栓也只能配对一个螺母。通过将螺母和螺栓组合在一起，木匠可以看出哪个更大，但木匠不能直接比较两个螺母或两个螺栓。

设计一种算法来解决这个问题，该算法使用 NlogN 次比较（以概率方式）。

Hint: *好好利用 Quicksort 中的划分过程。*

解答. 该问题的想法是利用快速排序中的划分过程，即利用螺母将螺栓进行划分，然后再利用螺栓将螺母进行划分，如此反复，直到螺母和螺栓都被划分完毕。记螺母集合为 L，螺栓集合为 I，算法流程如下：

(1) 随机选择一个螺母 x。
(2) 利用螺母将螺栓分成三部分：与其匹配的 N_x，比其大的螺栓集合 L_1，比其小的螺栓集合 L_2。
(3) 利用 N_x 将剩余的螺母分成两部分，比其大的螺母集合 I_1 和比其小的螺母集合 I_1。
(4) 分别在 (I_1, L_1) 和 (I_2, L_2) 中递归的调用上述过程，直到螺母和螺栓都被划分完毕。

粗略来看，每次划分的时间复杂性为 $O(n)$，而每次划分后，螺母和螺栓的数量都会减半，因此递归的深度为 $\log n$，从而算法的时间复杂性为 $O(n \log n)$.

我们下面给出一个准确的分析。事实上，我们只需要关注螺母的排序时间，因为每轮螺栓的操作个数与螺母的操作个数相同。设 $T(n)$ 为对 n 个螺母的排序时间，特别的我们令 L 中的螺母以升序排列 $x_1, \ldots, x_n$。我们考察 x_i 与 x_j 被比较次数。

- x_i 与 x_j 会被比较当且仅当 x_i 或者 x_j 被选为了划分的主元，从而其会被比较的概率 $p_{ij} = \frac{2}{j-i+1}$（这要求在 x_i 到 x_j 中的螺母中先选取这两个元素作为划分主元的概率）。

现在定义随机变量 X_{ij} 为：

$$X_{ij} = \begin{cases} 1, & \text{如果 } x_i \text{ 与 } x_j \text{ 被比较} \\ 0, & \text{其他情况} \end{cases}$$

则我们有算法对螺母的总比较次数为:

$$T(n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$$

计算其期望可得:

$$\begin{aligned}
E[T(n)] &= E[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}] \\
&= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}] \\
&= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \\
&= \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{2}{k} \\
&\approx 2n \ln n
\end{aligned}$$

从而算法的期望比较次数为 $O(n \log n)$. □

Remark 0.2

随机算法的分析需要利用到很多概率论的知识。事实上，上述也只是一个不太严谨的推理。仅供大家参考。但希望大家能认识到的是，随即挑选主元进行划分的快速排序算法的期望时间复杂性已经为 $O(n \log n)$.