Concrete Math: Homework 2

Due on March-23, 2022 at 14:00

Professor Chen Xue

SA21011018
Zhou Enshuai

2022 年 3 月 23 日

Problem 1

Modify QuickSort algorithm to find the k^{th} largest number: Given k and a sequence of n numbers $a1, ..., an$ (not sorted), output the k^{th} largest number.

1. Show the pseudo-code of your algorithm.
2. Prove its expected running time is $O(n)$ by the probability argument about comparing pairs of elements.

Solution

1.

Algorithm 1 Find the k^{th} largest number in an array

Input: A: a sequence of n numbers; *begin*: the index of the first number in A; *end*: the index of the last number in A; k: the ordinal number.

Output: the k^{th} largest number.

```
function FIND-KTH(A, begin, end, k)
    if k > (begin − end + 1) then
        return
    end if
    index = PARTITION(A, begin, end)
    if end - k + 1 < index then
        FIND-KTH(A, start, index - 1, k-(end - index + 1))
    else if end - k + 1 > index then
        FIND-KTH(A, index +1, end, k)
    else
        return A[index]
    end if
end function

function PARTITION(A, p, r)
    x = A[r]
    i = p-1
    for j = p to r-1 do
        if A[j] <= x then
            i = i+1
            swap(A[i], A[j])
        end if
    end for
    swap(A[i+1], A[r])
    return i+1
end function
```

2.

设为期望时间为 $T(n)$，已知调用 Partition 时间为 $O(n)$，最大比较次数为 n。设枢轴变量是第 l 大的元素，则 $p(l = i) = \frac{1}{n}$，$i = 1, 2, ..., n$，若 $l < k$，则搜索区间缩小为 $[l + 1, n]$，若 $l > k$，则搜索区间缩

小为 $[1, l-1]$，若 $l = k$，则搜索结束。则：

$$
\begin{aligned}
T(n) &\le \sum_{l=1}^{n} \frac{1}{n}\left[[l<k] \times T(n-l) + [l>k] \times T(l-1)\right] + n + 1 \\
&\le \sum_{l=1}^{n} \frac{1}{n}\left[[l<k] \times T(n-l) + [l>k] \times T(l-1) + [l==k] \times T(l-1)\right] + n + 1 \\
&\le \sum_{l=1}^{n} \frac{1}{n}\left[[l<k]T(max(n-l,l-1)) + [l>k]T(max(n-l,l-1)) + [l==k]T(max(n-l,l-1))\right] + n + 1 \\
&\le \frac{1}{n}\sum_{l=1}^{n} T(max(n-l,l-1)) + n + 1 \\
&\le \frac{1}{n}[\sum_{l=1}^{\lfloor\frac{n+1}{2}\rfloor} T(max(n-l,l-1)) + \sum_{l=\lceil\frac{n+1}{2}\rceil}^{n} T(max(n-l,l-1))] + n + 1 \\
&\le \frac{1}{n}[\sum_{l=1}^{\lfloor\frac{n+1}{2}\rfloor} T(n-l) + \sum_{l=\lceil\frac{n+1}{2}\rceil}^{n} T(l-1)] + n + 1 \\
&\le \frac{1}{n}[\sum_{l=\lfloor\frac{n}{2}\rfloor}^{n-1} T(l) + \sum_{l=\lceil\frac{n-1}{2}\rceil}^{n-1} T(l)] + n + 1 \\
&\le \frac{2}{n}\sum_{l=\lfloor\frac{n}{2}\rfloor}^{n-1} T(l) + n + 1
\end{aligned}
$$

可得：

$$
nT(n) \le 2\sum_{l=\lfloor\frac{n}{2}\rfloor}^{n-1} T(l) + n^2 + n
$$

由于求的是性能上界，不妨设等号成立：

$$
\begin{aligned}
nT(n) &= 2\sum_{l=\lfloor\frac{n}{2}\rfloor}^{n-1} T(l) + n^2 + n \\
(n-1)T(n-1) &= 2\sum_{l=\lfloor\frac{n-1}{2}\rfloor}^{n-2} T(l) + (n-1)^2 + n - 1
\end{aligned}
$$

两式相减可得：

$$
\begin{aligned}
&nT(n) - (n-1)T(n-1) = T(n-1) + 2n + T(\frac{n}{2}-1)[n\%2 == 0] \\
&nT(n) - (n-1)T(n-1) \le T(n-1) + 2n \\
&nT(n) \le nT(n-1) + 2n \\
&T(n) \le T(n-1) + 2 \\
&T(n) \le 2n
\end{aligned}
$$

所以，$T(n) = O(n)$

Problem 2

Prove that Euclid’s algorithm to compute $gcd(m, n)$ runs in time $O(logm + logn)$ assuming all integer-operations are done in $O(1)$ time.

Solution

设执行时间为 $T(m, n)$，显然 $T(m, n)$ 是关于 m, n 单调递增的。若 $m = n$，则复杂度为 $O(1)$，结论是平凡的；若 $n = 1$，则复杂度为 $T(m, 1) = O(1)$，结论也是平凡的；若 $m \neq n$，不妨设 $m > n$，可以分两种情况：

- 若 $m \geq 2n$，则 $m \mod n < n \leq \frac{m}{2}$
- 若 $m < 2n$，则 $m \mod n = m - n < m - \frac{m}{2} = \frac{m}{2}$

所以综上 $m \mod n \leq \frac{m}{2}$，则:

$$\begin{aligned}
T(m, n) &= 1 + T(n, m \mod n) \\
&= 1 + 1 + T(m \mod n, n \mod (m \mod n)) \\
&\leq 2 + T(\frac{m}{2}, \frac{n}{2}) \\
&\leq 2 \log m + 2 \log n + O(1) \\
&= O(\log m + \log n)
\end{aligned}$$

综上 $T(m, n) = O(\log m + \log n)$

Problem 3

1. 将 $\sum_{k=1}^{n} k2^k$ 重新改写成多重和式 $\sum_{1\le j\le k\le n} 2^k$ 的形式来对它进行计算
2. 用正文中的方法 5 来计算 $\text{⧉}_n = \sum_{k=1}^{n} k^3$：首先记 $\text{⧉}_n + \square_n = 2\sum_{1\le j\le k\le n} jk$ 然后应用（2.33）

Solution

1.

$$
\begin{aligned}
\sum_{k=1}^{n} k2^k &= \sum_{1\le j\le k\le n} 2^k \\
&= \sum_{j=1}^{n}\sum_{k=j}^{n} 2^k \\
&= \sum_{j=1}^{n}(2^{n+1} - 2^j) \\
&= (n-1)2^{n+1} + 2
\end{aligned}
$$

2.

$$
\begin{aligned}
\text{⧉}_n + \square_n &= \sum_{k=1}^{n}(k^3 + k^2) \\
&= \sum_{k=1}^{n}(\frac{k^2+k}{2} \times 2k) \\
&= \sum_{k=1}^{n}(\frac{k(k+1)}{2} \times 2k) \\
&= 2\sum_{k=1}^{n}(\sum_{j=1}^{k} j \times k) \\
&= 2\sum_{1\le j\le k\le n} jk \\
&= (\sum_{k=1}^{n} k)^2 + \sum_{k=1}^{n} k^2 \\
&= (\sum_{k=1}^{n} k)^2 + \square_n
\end{aligned}
$$

两边约去 $\square_n$，可得：

$$
\text{⧉}_n = (\sum_{k=1}^{n} k)^2 = \frac{n^2(n+1)^2}{4}
$$

Problem 4

1. 证明，表达式 $\lceil \frac{2x+1}{2} \rceil - \lceil \frac{2x+1}{4} \rceil + \lfloor \frac{2x+1}{4} \rfloor$ 总是等于 $\lfloor x \rfloor$ 或者 $\lceil x \rceil$
2. 证明序列 $1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5,$ 的第 n 个元素是 $\lfloor \sqrt{2n} + \frac{1}{2} \rfloor$

Solution

1.

对于任意整数 n 显然有：

$$\lceil n \rceil - \lfloor n \rfloor = 0$$

对于任意非整数 y 显然有：

$$\lceil y \rceil - \lfloor y \rfloor = 1$$

所以当 $(2x+1) \mod 4 = 0$，也即 $x = 2n + \frac{3}{2}$，$n \in Z$ 时：

$$\lceil \frac{2x+1}{4} \rceil - \lfloor \frac{2x+1}{4} \rfloor = 0$$

否则：

$$\lceil \frac{2x+1}{4} \rceil - \lfloor \frac{2x+1}{4} \rfloor = 1$$

可得：

$$\begin{aligned}
\lceil \frac{2x+1}{2} \rceil - \lceil \frac{2x+1}{4} \rceil + \lfloor \frac{2x+1}{4} \rfloor &= \lceil x + \frac{1}{2} \rceil - [x \neq 2n + \frac{3}{2}] \\
&= \lceil x \rceil + [\{x\} > \frac{1}{2}] + [\{x\} = 0] - [x \neq 2n + \frac{3}{2}] \\
&= \begin{cases} \lceil x \rceil, \ \{x\} > \frac{1}{2} \\ \lceil x \rceil, \ \{x\} \leq \frac{1}{2}, \ (2x+1)\%4 = 0 \\ \lfloor x \rfloor, \ 0 < \{x\} \leq \frac{1}{2}, \ (2x+1)\%4 \neq 0 \\ \lfloor x \rfloor, \ \{x\} = 0, \ (2x+1)\%4 \neq 0 \end{cases}
\end{aligned}$$

2.

设第 n 个元素为 a_n，当 $a_n = m$ 时，前面小于 m 的数 $1, 2, 2, ..., m-1$ 的个数是：

$$\sum_{i=1}^{m-1} = \frac{m(m-1)}{2}$$

值为 m 的数组元素的个数是 m，所以值为 m 的数组元素 a_n 的索引 n 的范围是：

$$\begin{aligned}
\frac{m(m-1)}{2} &< n \leq \frac{m(m-1)}{2} + m \\
m^2 - m &< 2n \leq m^2 + m \\
m^2 - m + \frac{1}{4} &< 2n \leq m^2 + m + \frac{1}{4} \\
(m - \frac{1}{2})^2 &< 2n \leq (m + \frac{1}{2})^2 \\
m - \frac{1}{2} &< \sqrt{2n} \leq m + \frac{1}{2}
\end{aligned}$$

可得 m 的范围：

$$\sqrt{2n} - \frac{1}{2} \leq m < \sqrt{2n} + \frac{1}{2} \tag{1}$$

Problem 4 continued on next page...

假设 $\sqrt{2n}-\frac{1}{2}$ 是整数，即 $\sqrt{2n}-\frac{1}{2}=k\in Z$：

$$2n=(k+\frac{1}{2})^2$$
$$2n=k^2+k+\frac{1}{4}$$
$$2n-k(k+1)=\frac{1}{4}$$

等式左边是整数，右边不是整数，显然推出矛盾，故 $\sqrt{2n}-\frac{1}{2}$ 不是整数，那么 $\sqrt{2n}-\frac{1}{2}+1=\sqrt{2n}+\frac{1}{2}$ 也不是整数，所以：

$$\left\lceil\sqrt{2n}-\frac{1}{2}\right\rceil=\left\lfloor\sqrt{2n}-\frac{1}{2}+1\right\rfloor=\left\lfloor\sqrt{2n}+\frac{1}{2}\right\rfloor$$

再带入不等式 (1) 可得：

$$\sqrt{2n}-\frac{1}{2}\le m\le\left\lfloor\sqrt{2n}+\frac{1}{2}\right\rfloor$$
$$\left\lceil\sqrt{2n}-\frac{1}{2}\right\rceil\le m\le\left\lfloor\sqrt{2n}+\frac{1}{2}\right\rfloor$$
$$\left\lfloor\sqrt{2n}+\frac{1}{2}\right\rfloor\le m\le\left\lfloor\sqrt{2n}+\frac{1}{2}\right\rfloor$$

故：

$$a_n=m=\left\lfloor\sqrt{2n}+\frac{1}{2}\right\rfloor$$

Problem 5

求解递归式：

$$\begin{cases} a_0 = 1; \\ a_n = a_{n-1} + \left\lfloor \sqrt{a_{n-1}} \right\rfloor, \ n > 0. \end{cases} \tag{2}$$

Solution

当 $a_n = m^2$ 时：

$$\begin{aligned} a_{n+1} &= m^2 + m, & a_{n+2} &= m^2 + m + m = m^2 + 2m \\ a_{n+3} &= m^2 + 2m + m = (m+1)^2 + (m-1), & a_{n+4} &= m^2 + 3m + m + 1 = (m+1)^2 + 2m \\ a_{n+5} &= m^2 + 5m + 2 = (m+2)^2 + (m-2), & a_{n+6} &= m^2 + 3m + m + 1 = (m+2)^2 + 2m \end{aligned} \tag{3}$$

引理：

$$a_{n+2k+1} = (m+k)^2 + (m-k), \ a_{n+2k+2} = (m+k)^2 + 2m, \ 0 \le k \le m \tag{4}$$

用数学归纳法证明引理（4）：

- 当 $k = 0$ 时，$a_{n+2k+1} = a_{n+1} = m^2 + m$，$a_{n+2k+2} = a_{n+2} = m^2 + 2m$，引理（4）成立
- 假设 $k = l, \ l < m$ 时引理（4）成立，当 $k = l + 1$ 时：

$$\begin{aligned} a_{n+2(l+1)+1} &= a_{n+2l+3} = (m+l)^2 + 2m + m + l = (m + (l+1))^2 + m - (l+1) \\ a_{n+2(l+1)+2} &= a_{n+2l+4} = (m + (l+1))^2 + m - (l+1) + m + (l+1) = (m + (l+1))^2 + 2m \end{aligned}$$

综上引理（4）成立。令 $k = m$，可得：

$$a_{n+2m+1} = (m+m)^2 + (m-m) = (2m)^2$$

此时又得到一个完全平方数。这样一来，我们可以发现，以某一完全平方数 $a_n = m^2$ 为起点，可以利用引理（4）计算出其后面的 $2m+1$ 个数，即：$a_n, a_{n+1}, .., a_{n+2m+1}$，此时引理（4）已不能继续递推，需要重新更换起点，该起点得是完全平方数，而 $a_{n+2m+1} = 2m^2$ 刚好是完全平方数，以其为起点，又可以继续下一轮递推。

由于 $a_0 = 1$ 是完全平方数，因此，可以以 $a_0 = 1$ 为起点递推, 得到：a_0, a_1, a_2, a_3；再以 a_3 为起点，继续递推，得到：$a_3, a_4, a_5, .., a_8$；再以 a_8 为起点，继续递推，得到：$a_8, a_9, a_10, .., a_{17}$；……；

除去左起始端点，每轮递推依次得到 $2^1+1, 2^2+1, 2^3+1, ..., 2^m+1, ...$ 个数。

左起始端点依次为 $a_{2^1+1-3}, a_{2^2+2-3}, a_{2^3+3-3}, .., a_{2^m+m-3}, ...$，所以，当 $2^l + l - 3 \le n < 2^{l+1} + l - 2$ 时：

$$a_n = 2^l + \left\lfloor \left(\frac{n-l-1}{2} \right)^2 \right\rfloor$$