问题求解（二）作业（第九周）

161180162 许致明

2018 年 5 月 13 日

TC 第七章

7.1-3

证明. **for** 循环共执行了 $r-p$ 次，而每次执行消耗 $\Theta(1)$ 时间，而循环外的操作消耗时间也为 $\Theta(1)$。因此 Partition 消耗 $\Theta(r-p)$ 时间，即 $\Theta(n)$，n 为子数组的大小。 □

7.2-4

证明. 假定该数组中有 c 个元素是未排列好的，c 为与数组大小 n 无关的常数。对于插入排序，循环次数为 c，则运行时间为 $O(cn)=O(n)$。而对快速排序，这种输入将会产生最坏情况划分，导致最坏运行时间 $O(n^2)$。因此对于此类输入，插入排序效果好于快速排序。 □

7.3-2

最坏情况：$\Theta(n)$
最好情况：$\Theta(n)$

7.4-2

设最好情况运行时间为 $T(n)$，则：

$$T(n)=\min_{1\le q\le n-1}\left(T(q)+T(n-q-1)\right)+O(n)$$

假设 $T(n)\ge c\left(n\log n+2n\right)$，$c$ 为常数

$$\begin{aligned}
T(n)&\ge \min_{1\le q\le n-1}\left(cq\log q+2cq+c\left(n-q-1\right)\log\left(n-q-1\right)+2c\left(n-q-1\right)\right)+\Theta(n)\\
&=\frac{cn}{2}\log(n/2)+cn+c(n/2-1)\log(n/2-1)+cn-2c+\Theta(n)\\
&\ge cn\log n+cn/2-\log n+2-2c+\Theta(n)
\end{aligned}$$

当 $q=n/2$ 时，上式最小，得 $T(n)\ge cn\log n$，即 $T(n)=\Omega(n\log n)$

7-5

a.
$$p_i = \frac{6(n-i)(i-1)}{n(n-1)(n-2)}$$

b.
$$i = \left\lfloor \frac{n+1}{2} \right\rfloor$$
$$\text{则结果为：} f(n,i) = \frac{6\left(\left\lfloor \frac{n-1}{2} \right\rfloor\right)\left(n - \left\lfloor \frac{n+1}{2} \right\rfloor\right)}{n(n-1)(n-2)} - \frac{1}{n}$$
$$\lim_{n\to\infty} f(n,i) = \frac{3}{2}$$

c. 设 n 为 3 的幂，则：
$$S(n) = \sum_{i=n/3}^{2n/3} p_i \approx \int_{n/3}^{2n/3} \frac{6(-x^2+nx+x-n)}{n(n-1)(n-2)}\,\mathrm{d}x$$
$$\lim_{n\to\infty} S(n) = \frac{13}{27} > \frac{1}{3}$$

d. 三数取中法能得到更好的划分。但是根据 7.4 节内容，任何常数比例的划分都会产生 $\Theta(\log n)$ 的递归树，且每层的代价为 $\Theta(n)$。因此仍有 $\Omega(n\log n)$ 的复杂度下界，但更好的划分可以减小隐藏在 Ω 符号中的常数。

TC 第八章

8.1-4

对于整个输入序列，共有 $k!^{n/k}$ 种输出的可能，设搜索树的高度为 h，则：

$$k!^{n/k} \le 2^h$$
$$h \ge (n/k)\log(k!) \ge (n/k)\left(\frac{k\ln k}{\ln 2}\right) = \Omega(n\log k)$$

8.2-4

执行计数排序的 1-9 行，然后返回 $C[b]-C[a-1]$。

8.3-4

将所有数字转换为 n 进制数，然后使用基数排序。

8.2-4

当所有元素都在同一个桶中出现最坏情况，这时需要 $O(n^2)$ 的运行时间。解决方法：对每个桶都使用归并排序，这样最坏复杂度为 $O(n\log n)$

8-2

a.

```
SORT-A(A)
Let C be a new array
index=1
for i == 1 to n
    if A[i]==0
        C[index]=A[i]
        index=index+1
for i == 1 to n
    if A[i]==1
        C[index]=A[i]
        index=index+1
return C
```

b.

```
SORT-B(A)
index=1
for i=1 to n
    if A[i]==0
        SWAP(A[i],A[index])
        index=index+1
return A
```

c. 冒泡排序

d. 使用 SORT-A

e. 先运行计数排序的 $1-5$ 行，然后将得到的数字复制一份，运行 $7-9$ 行

9.1-1

(1) $n-1$ 次比较找到最小值

(2) 在大于最小值的 $\lceil \log n \rceil$ 个数中找到次小值，比较 $\lceil \log n \rceil - 1$ 次

比较次数 $n + \lceil \log n \rceil - 2$

9.3-7

(1) 找到中位数

(2) 每个数减去中位数，取绝对值

(3) 找到第 k 小的数

(4) 选取绝对值小于等于第 k 小的数的所有数