7.Quicksort

Hsu, Lih-Hsing

7.1 Description of quicksort

- *Divide*
- *Conquer*
- *Combine*

QUICKSORT(A,p,r)

if $p < r$

then $q \leftarrow PARTITION(A, p, r)$

QUICKSORT(A,p,q)

QUICKSORT($A,q+1,r$)

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

      exchange A[i] ↔ A[j]

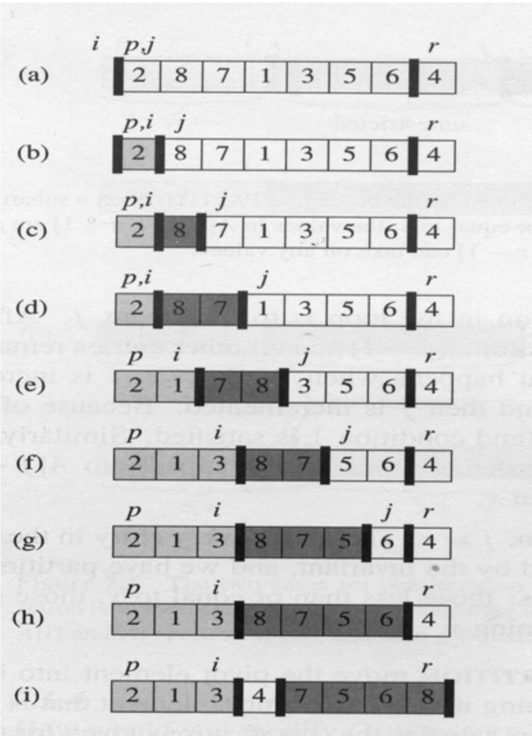
exchange A[i +1] ↔ A[r]

return i +1

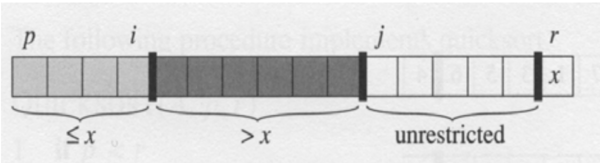
At the beginning of each iteration of the loop of lines 3-6, for any array index k,

1. if $p \leq k \leq i$, then $A[k] \leq x$.
2. if $i + 1 \leq k \leq j - 1$, then $A[k] > x$.
3. if $k = r$, then $A[k] = x$.

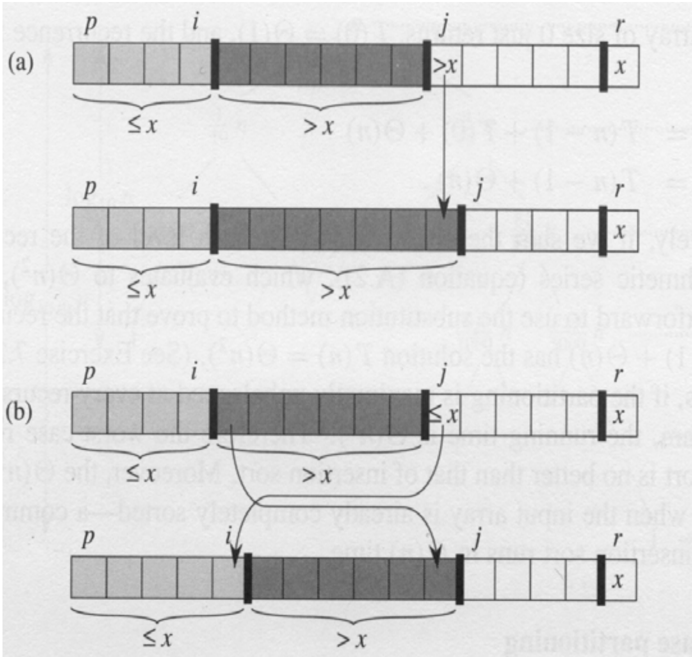
The operation of *Partition* on a sample array



Two cases for one iteration of procedure *Partition*



Complexity:
Partition on A[p…r] is Θ(n)
where n = r –p +1



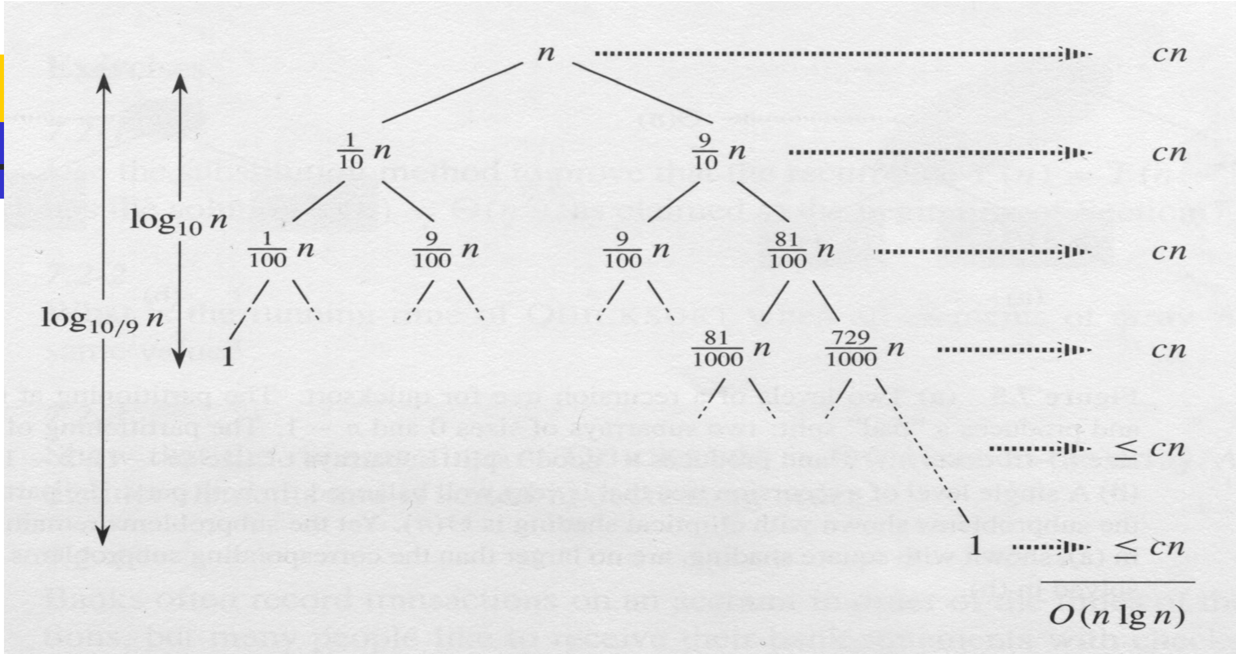
7.2 Performance of quicksort

Worst-case partition:

$$T(n) = T(n-1) + \Theta(n)$$
$$= \sum_{k=1}^{n} \Theta(k) = \Theta\left(\sum_{k=1}^{n} k\right) = \Theta(n^2)$$

Best-case partition:

$$T(n) = 2T(n/2) + \Theta(n)$$
$$\Rightarrow T(n) = \Theta(n \log n)$$

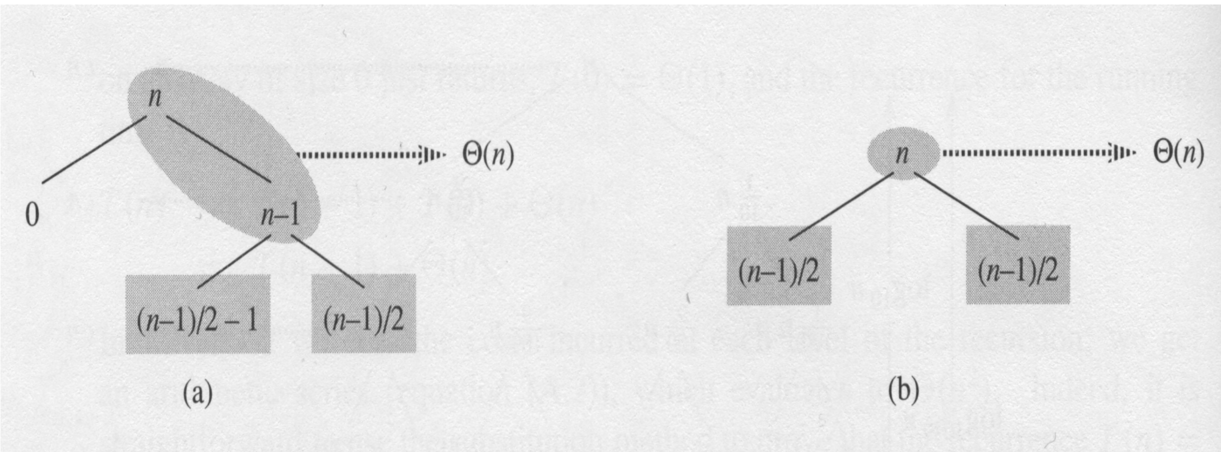


Balanced partition $T(n) = \Theta(n \log n)$

$$T(n) = T(9n/10) + T(n/10) + \Theta(n)$$

$$\Rightarrow T(n) = \Theta(n \log n)$$

Intuition for the average case T(n) = Θ(*n* log*n*)



7.3 Randomized versions of partition

RANDOMIZED_PARTITION(A,p,r)

1 $i \leftarrow RANDOM(p,r)$

2 exchange $A[p] \leftrightarrow A[i]$

3 **return** PARTITION(A,p,r)

RANDOMIZED_QUICKSORT(A,p,r)

1 **if** $p < r$

2 **then**

$q \leftarrow RANDOMIZED\_PARTITION(A,p,r)$

3 RANDOMIZED_QUICKSORT(*A*,*p*,*q*)

4 RANDOMIZED_QUICKSORT(*A*,*q*+1,*r*)

7.4 Analysis of quicksort

7.4.1 Worst-case analysis

$$T(n) = \max_{0 \le q \le n-1} (T(q) + T(n-q-1)) + \Theta(n)$$

guess $T(n) \le cn^2$

$$\begin{aligned} T(n) &\le \max_{0 \le q \le n-1} (cq^2 + c(n-q-1)^2) + \Theta(n) \\ &= c \max_{0 \le q \le n-1} (q^2 + (n-q-1)^2) + \Theta(n) \\ &\le cn^2 - 2c(n-1) + \Theta(n) \\ &\le cn^2 \end{aligned}$$

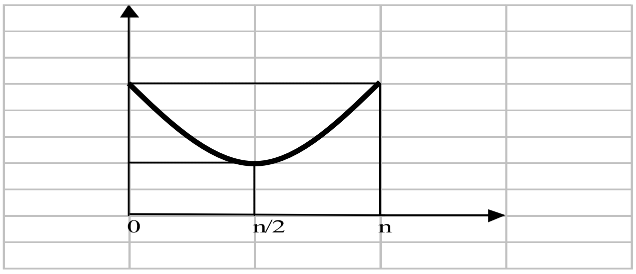
pick the constant c large enough so that the $2c(n-1)$ term dominates the $\Theta(n)$ term.

$\Rightarrow T(n) = \Theta(n^2)$

7.4-2

Show that $q^2+(n-q)^2$ **achieves a maximum over**

$q=1,2,\ldots\ldots,n-1$ **when** $q=1$ **or** $q=n-1$



ans: 先令 $f(q)=q^2+(n-q)^2$

一次微分: $f'(q)=2q-2(n-q)=4q-2n$

令 $f'(q)=0 \Rightarrow 4q-2n=0 \Rightarrow q=\frac{n}{2}$(極小值)

二次微分: $f''(q)=4$ (開口向上)

因為 $1\le q\le n-1$ 所以 $f(1)=f(n-1)=1+(n-1)^2$ (相對極大值)

7.4.2 Expected running time

- Running time and comparsions
- Lemma 7.1
  - Let X be the number of comparisons performed in line 4 of *partition* over the entire execution of *Quicksort* on an *n*-element array. Then the running rime of *Quicksort* is $O(n+X)$

we define

$$X_{ij} = I\ \{z_i \text{ is compared to } z_j\},$$

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}.$$

$$E[X] = E\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}\right]$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E\left[X_{ij}\right]$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Pr\ \{z_i \text{ is compared to } z_j\}$$

$$\begin{aligned}\Pr\{z_i \text{ is compared to } z_j\} &= \Pr\{z_i \text{ or } z_j \text{ is first pivot chosen from } Z_{ij}\}\\ &= \Pr\{z_i \text{ is first pivot chosen from } Z_{ij}\}\\ &\quad + \Pr\{z_j \text{ is first pivot chosen from } Z_{ij}\}\\ &= \frac{1}{j-i+1} + \frac{1}{j-i+1}\\ &= \frac{2}{j-i+1}\end{aligned}$$

$$\therefore E[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1}.$$

$$E[X]=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{2}{j-i+1}$$

$$=\sum_{i=1}^{n-1}\sum_{k=1}^{n-i}\frac{2}{k+1}$$

$$<\sum_{i=1}^{n-1}\sum_{k=1}^{n}\frac{2}{k}$$

$$=\sum_{i=1}^{n-1}O(\lg n)$$

$$=O(n\lg n)$$

another analysis

$$T(n)=$$

$$\frac{1}{n}(T(1)+T(n-1)+\sum_{q=1}^{n-1}(T(q)+T(n-q))+\Theta(n)$$

$$\left.\begin{array}{l}T(1)=1\\ T(n-1)=O(n^2)\end{array}\right\}\Rightarrow\frac{1}{n}(T(1)+T(n-1))=O(n)$$

$$T(n)=$$

$$\frac{1}{n}(T(1)+T(n-1)+\sum_{q=1}^{n-1}(T(q)+T(n-q))+\Theta(n)$$

$$=\frac{1}{n}(\sum_{q=1}^{n-1}T(q)+T(n-q))+\Theta(n)$$

$$=\frac{2}{n}(\sum_{k=1}^{n-1}T(k))+\Theta(n)$$

guess $T(n) \le an\log n + b$

$$T(n) \le \frac{2}{n}\left(\sum_{k=1}^{n-1} ak\log k + b\right) + \Theta(n)$$

$$= \frac{2a}{n}\sum_{k=1}^{n-1} k\log k + \frac{2b}{n}(n-1) + \Theta(n)$$

We will prove $\sum_{k=1}^{n-1} k\log k \le \frac{n^2\log n}{2} - \frac{n^2}{8}$

$$T(n) \le \frac{2a}{n}\left(\frac{1}{2}n^2\log n - \frac{n^2}{8}\right) + \frac{2b(n-1)}{n} + \Theta(n)$$

$$\le an\log n - \frac{an}{4} + 2b + \Theta(n)$$

$$= an\log n + b + \left(\Theta(n) + b - \frac{an}{4}\right)$$

$$\le an\log n + b$$

Choose a large enough so that $\frac{an}{4} \ge \Theta(n) + b$.

$\Rightarrow T(n) = O(n\log n)$.

$$\sum_{k=1}^{n-1} k\log k = \sum_{k=1}^{\lceil n/2\rceil-1} k\log k + \sum_{k=\lceil n/2\rceil}^{n-1} k\log k$$

$$\le (\log n - 1)\sum_{k=1}^{\lceil n/2\rceil-1} k + \log n \sum_{k=\lceil n/2\rceil}^{n-1} k$$

$$= \log n \sum_{k=1}^{n-1} k - \sum_{k=1}^{\lceil n/2\rceil-1} k$$

$$\le \frac{n(n-1)\log n}{2} - \frac{1}{2}\left(\frac{n}{2}-1\right)\frac{n}{2}$$

$$\le \frac{n^2\log n}{2} - \frac{n^2}{8}$$

if $n \ge 2$.

Another approach: Using

$$\int x \ln x dx = \frac{1}{2}x^2 \ln x - \frac{1}{4}x^2$$