------------------------------ 7 ------------------------------

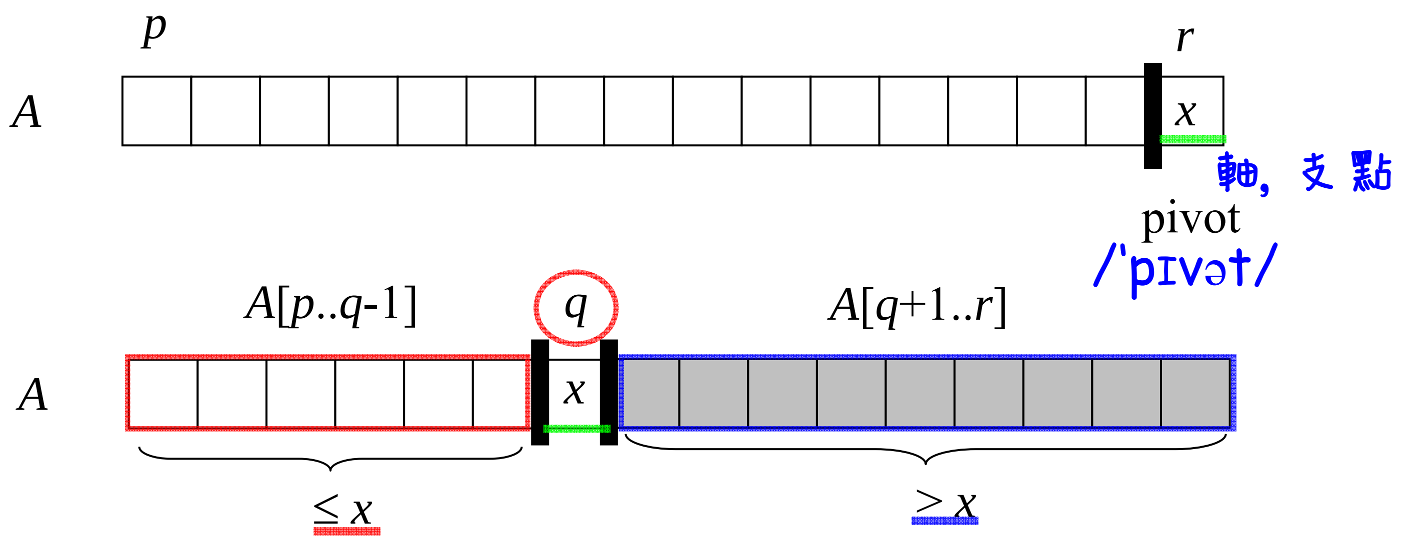
Quicksort

7-1x

7.1 Quicksort

Quicksort(A[p..r])

Divide: partition $A[p..r]$ into $A[p..q-1]$ and $A[q+1..r]$



Conquer: recursively sort $A[p..q-1]$ and $A[q+1..r]$
Combine: no work is needed.

Quicksort(A, p, r)

* p < r ➔ 至少 2 個 (0 or 1 個 不

If $p < r$ **then**
    $q \leftarrow$ Partition(A, p, r)   /* **divide** */
    Quicksort(A, p, q-1)   /* **conquer** */
    Quicksort(A, q+1, r)   /* **conquer** */

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
```

i+1: 是下一個紅色的位置

☆ i+1: 是下一個紅色的位置 = 1st 藍色位置

= q

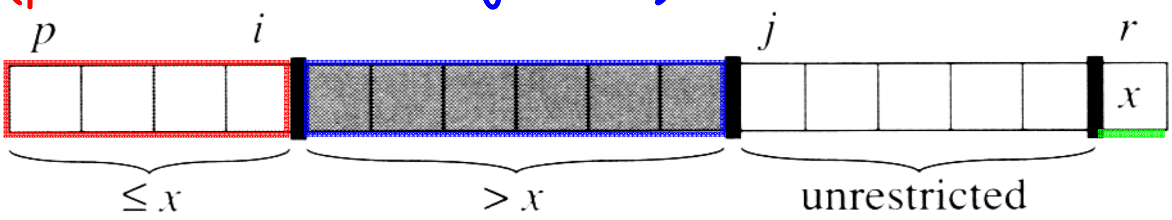
i+1: 1st 藍色位置

j: 下一個要檢查 (p ~ r-1)

i: 目前最後一個紅色位置

Meaning of *i* and *j*:

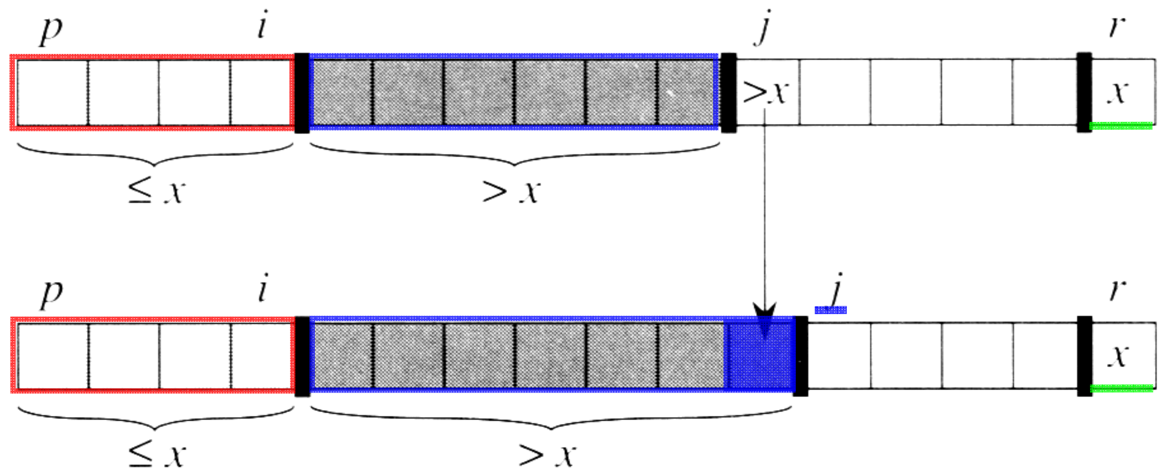
(p ~ i: 紅色， i+1 ~ j-1 : 藍)



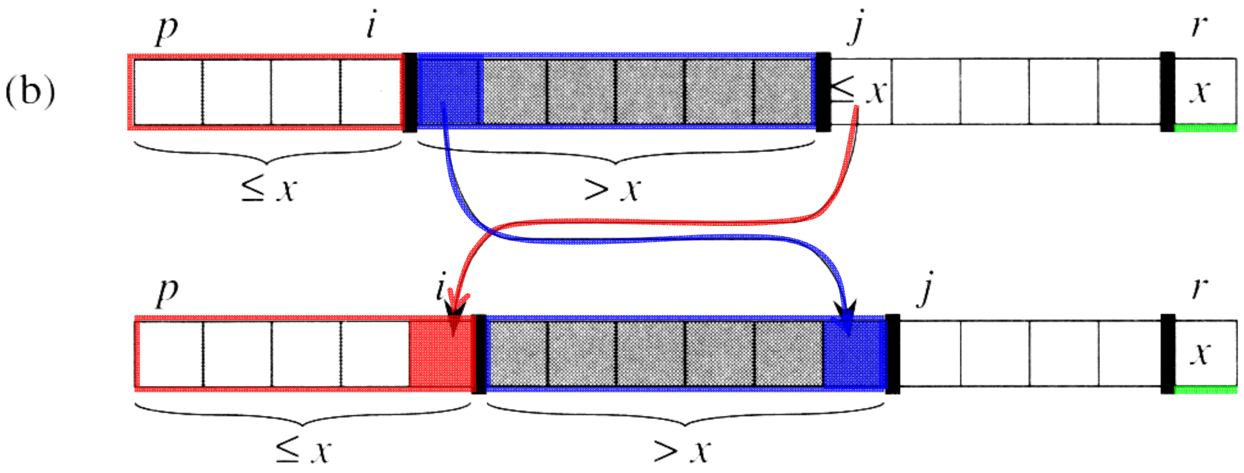
How *i* and *j* changes:

case 1. A[j] > x

(a)



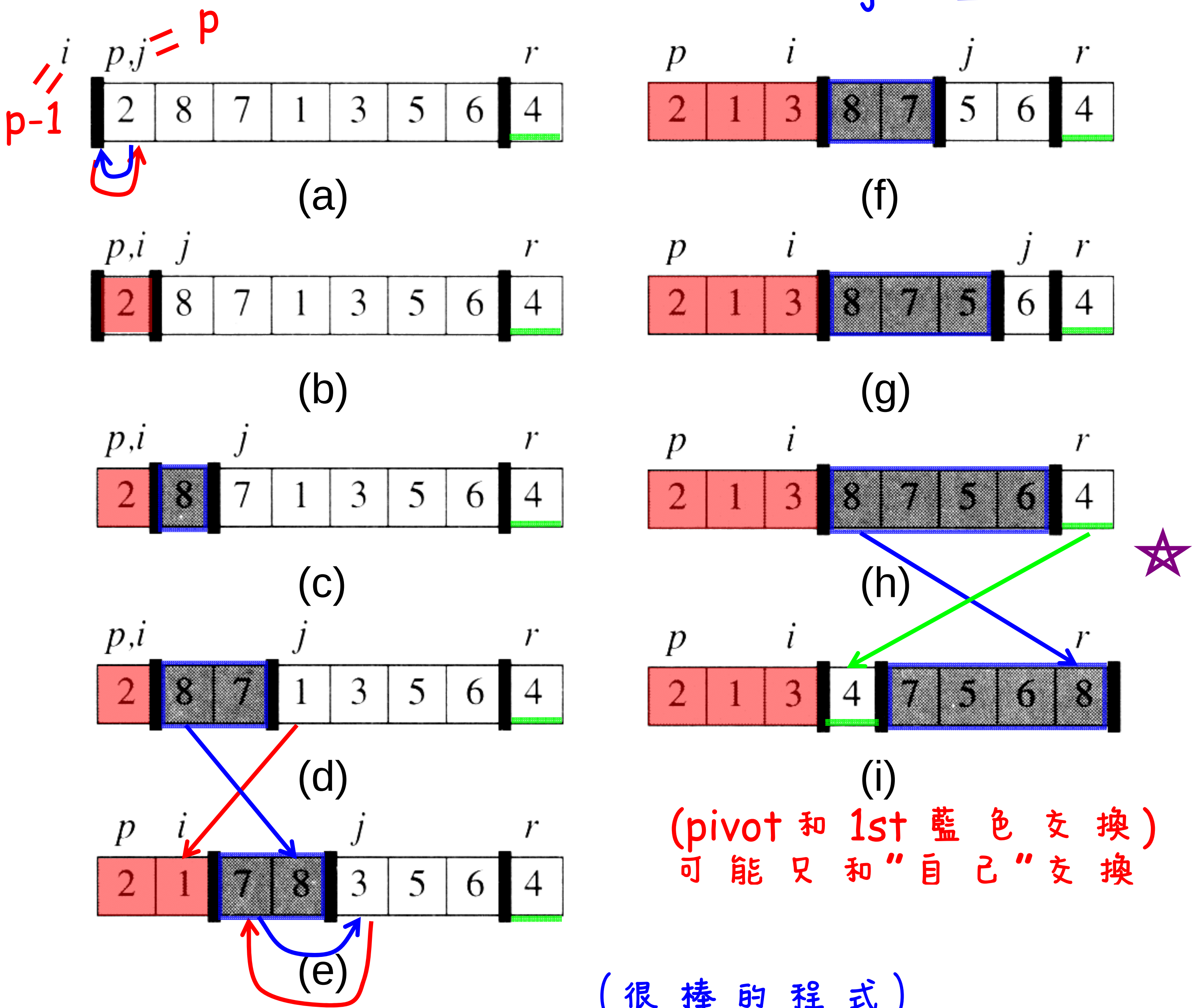
case 2. $A[j] \le x$



Example: (Partition, $x=A[r]=4$)

p ~ i: 紅色

i+1 ~ j-1: 藍色



(pivot 和 1st 藍色交換)

可能只和"自己"交換

(很棒的程式)

* 開始與過程中沒有紅，也沒有藍都要會對！
* 最後沒藍也會對！

7.4 Analysis

for example

Worst-case: $\Theta(n^2)$ (occurs for sorted input)

7-4x

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

(or simply n, n-1)

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

* $T(n) = O(n^2)$

Guessing: $T(n) \le c\, n^2 = O(n^2)$ (for $n >= n_0$)

Substituting: (Basis: ok)

(Induction) Assume $T(x) \le cx^2$ for $x = n_0, \ldots, n-1$

$$T(n) \le \max_{0 \le k \le n-1} \{ck^2 + c(n-k-1)^2\} + \Theta(n)$$

$$\le c \max_{0 \le k \le n-1} \{k^2 + (n-k-1)^2\} + \Theta(n)$$

$$\le c(n-1)^2 + \Theta(n)$$

$$\le cn^2 - c(2n-1) + \Theta(n)$$

e.g. 100n

$\Theta(n) = bn$
b is a constant

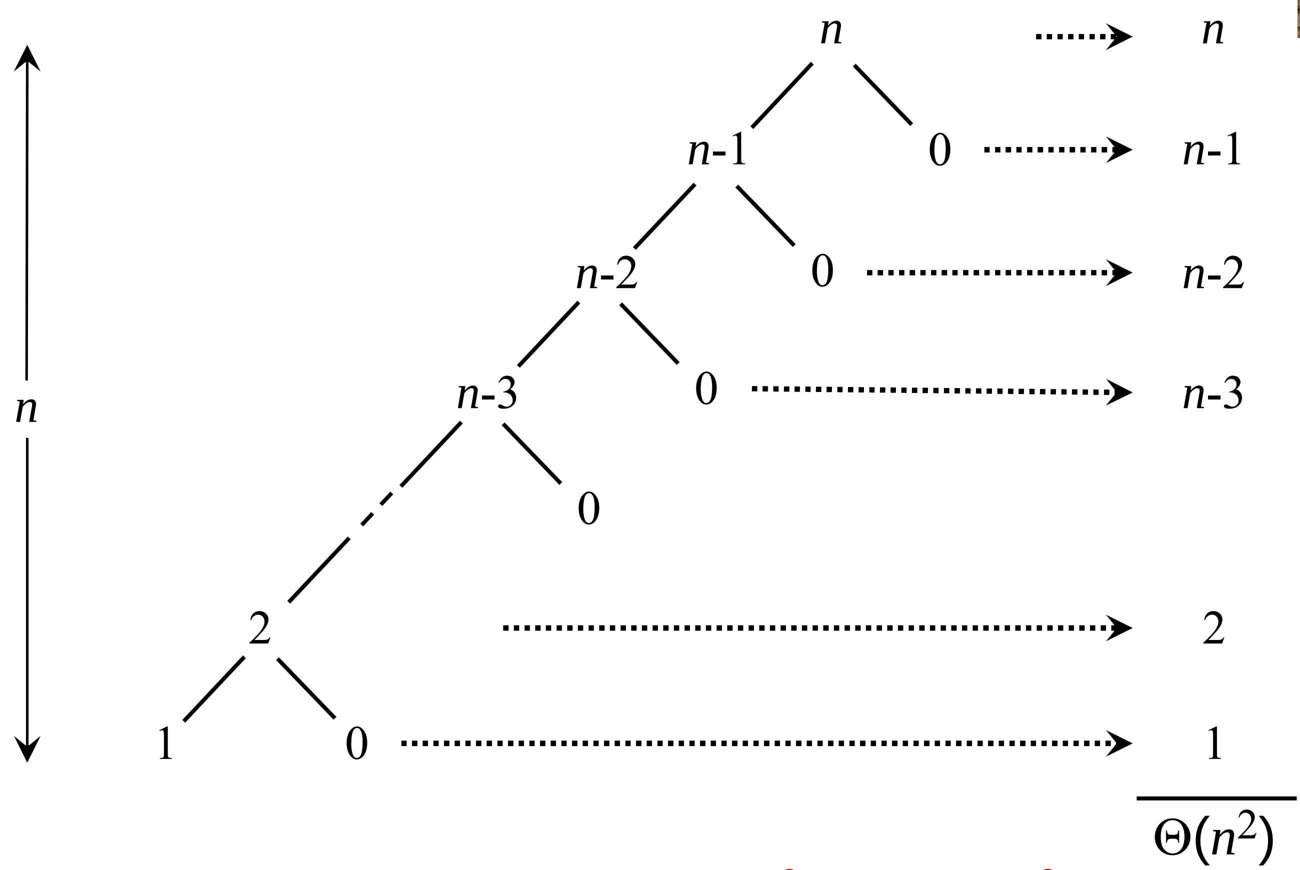
$$\le cn^2$$ (by picking c large enough)

goal!

- $T(n) = \Omega(n^2)$

Worst-case partition ? (proof?)
(e.g., occurs for sorted input)

7-5x



* This example shows $T(n) = \Omega(n^2)$, not $O(n^2)$

Average-case: $\Theta(n \lg n)$

(Assume that all elements are distinct.)

→ at most $O(n)$ calls to Quicksort

7-5a

- $T(n)=O(n+X)$, where X is the total # of comparisons performed in Line 4 of Partition.
- For each call to Partition, $A[i]$ and $A[j]$ will not be further compared if $A[i]<x<A[j]$ or $A[j]<x<A[i]$.

Example: Let A={3, 9, 2, 7, 5}. After 1st round, A={3, 2, 5, 9, 7}. Then, {3, 2} won't be compared with {9, 7} anymore.

7-5b

- Rename the elements of A as $z_1, z_2, \ldots, z_n$ with z_i being the i-th smallest element. Also define the set $Z_{ij}=\{z_i, z_{i+1}, \ldots, z_j\}$.

- $z_i : z_j$ iff the first pivot chosen from Z_{ij} is either z_i or z_j.

- For any i,j, the probability of $z_i : z_j$ is $2/(j-i+1)$. Therefore,

$$X = \overbrace{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}}^{\sum_{i,j\,:\,i<j}} \frac{2}{j-i+1} \times 1 \text{ comparison}$$

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

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

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

(Using Harmonic Series)

$$H_n = \sum_{k=1}^{n} \frac{1}{k} = \ln n + O(1)$$

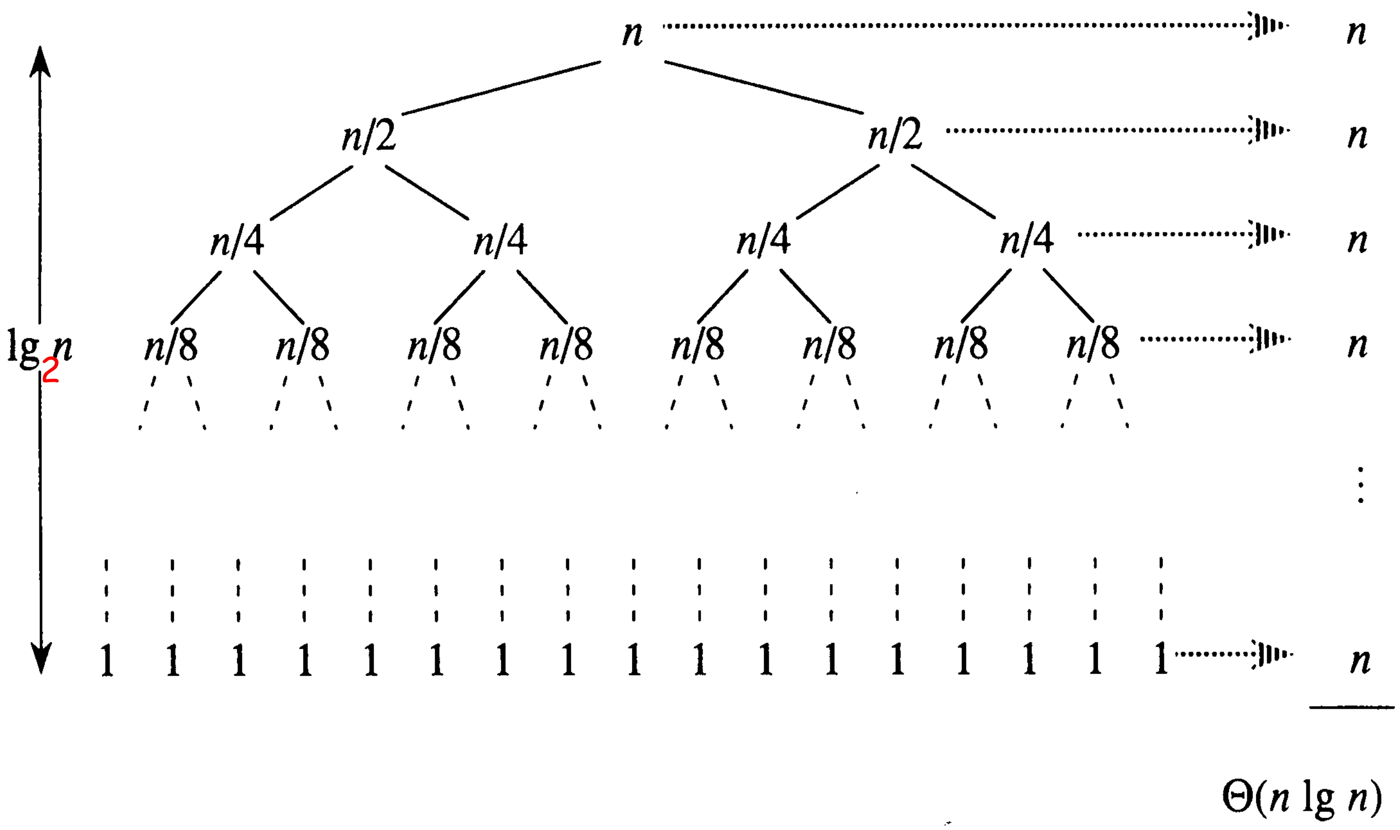
$$= O(\lg n)$$

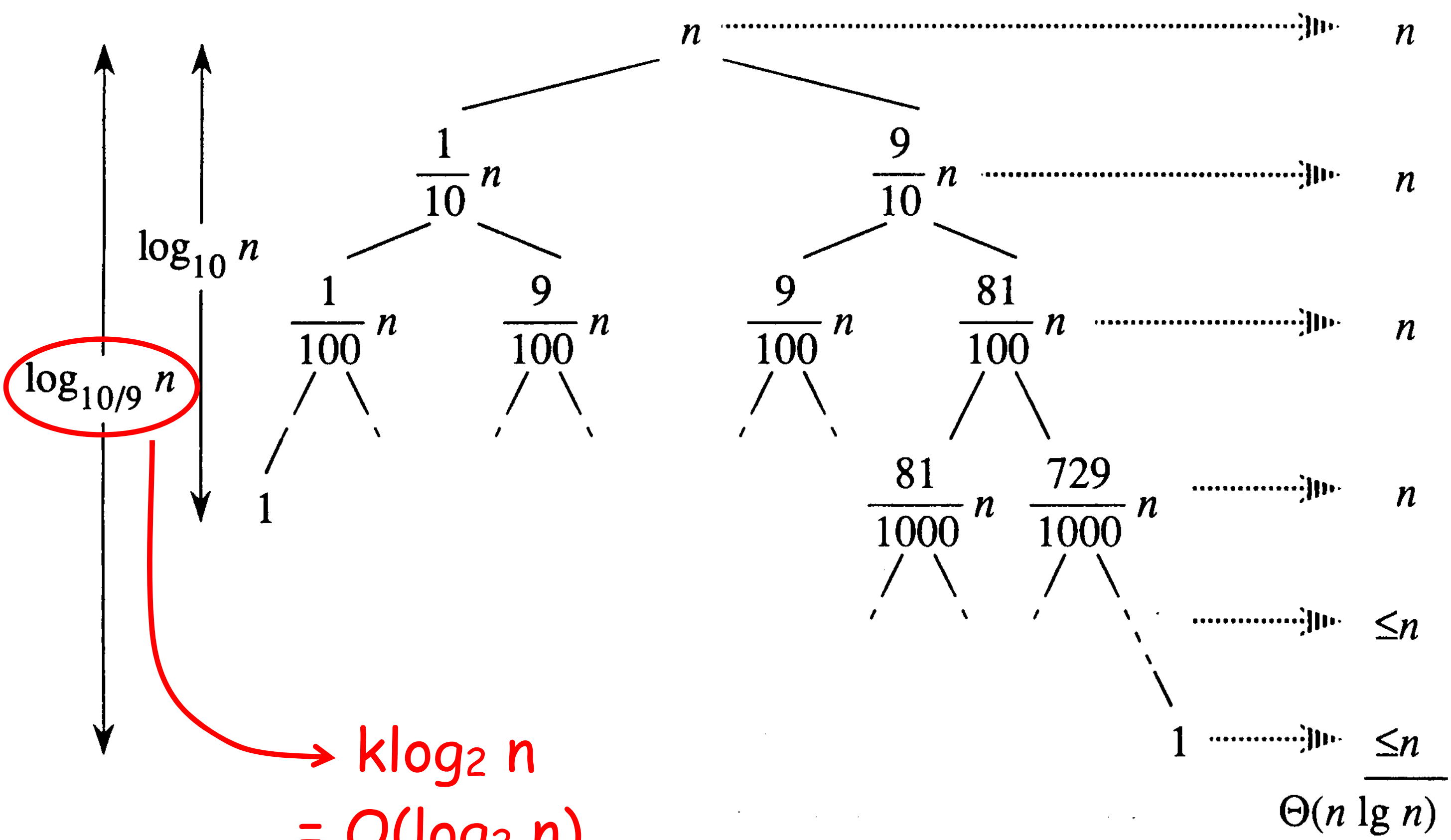
(see 3-5)

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

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

best-case partition ? (proof?)





$k \log_2 n$

$= O(\log_2 n)$

7-8a
7-8b

- **Another analysis**

n-1 ⋮ 2 1 0   0 1 2 ⋮ n-1

$$E(n) = \underline{(n-1)} + \frac{1}{n}\sum_{q=1}^{n}\{E(q-1)+E(n-q)\}$$

only # of comparisons is counted

$$= \underbrace{(n-1)}_{n,\,bn,\,\Theta(n)} + \frac{2}{n}\sum_{k=1}^{n-1}E(k)$$

For simplicity, assume

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

$$\Rightarrow nE(n) = n^2+n + 2\sum_{k=1}^{n-1}E(k) \qquad \text{------(1)}$$

$$\Rightarrow (n\text{-}1)E(n\text{-}1) = (n-1)^2+(n-1)+2\sum_{k=1}^{n-2}E(k) \qquad \text{------(2)}$$

(replacing n by n-1 in (1))

(1)–(2), we have

$$nE(n) = (n+1)E(n-1)+2n$$

$$\Rightarrow \quad E(n) = \frac{n+1}{n}E(n-1)+2$$

(Applying iteration method)

$$E(x) = \frac{x+1}{x}E(x-1)$$

$$= \frac{n+1}{n}\left\{\frac{n}{n-1}E(n-2)+2\right\}+2$$

$$= \frac{n+1}{n-1}E(n-2)+2\frac{n+1}{n}+2$$

$$= \frac{n+1}{n-2}E(n-3)+2\frac{n+1}{n-1}+2\frac{n+1}{n}+2$$

$$= \bullet \quad \bullet \quad \bullet \quad \bullet$$

$$= \frac{n+1}{2}E(1)+2(n+1)\left(\frac{1}{3}+\frac{1}{4}+\ldots+\frac{1}{n}\right)+2$$

$$= \Theta(n)+\Theta(n)\sum_{k=3}^{n}\frac{1}{k}+2$$

$$= \Theta(n)+\Theta(n)\times\Theta(\lg n)+2 \quad \text{(Using Harmonic Series)}$$

$$= \Theta(n \lg n)$$

7.3 Randomized version of quicksort

Randomized Algorithm:

An algorithm uses *random-number* generator.

Pseudorandom-number generator:

A deterministic algorithm that returns numbers that "look" statistically random.

Randomized-Partition(A, p, r)

$i \leftarrow$ Random(p, r); pivot

exchange($A[r]$, $A[i]$); prevent almost sorted or attack

return Partition(A, p, r)

Homework: Ex. 7.1-2, 7.2-4, 7.4-2 and Pro. 7-1, 7-4

7-10a

original partition

保證 problem-size 會減小!

經典之作! 請勿任意更動!

* need a q-sort:

copy one, instead of write one by yourself

(自己寫的一定不是"quick" sort)