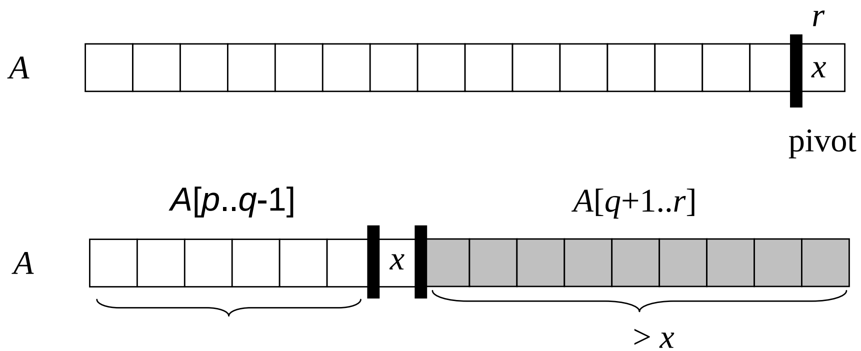
# ------------------------------ 7 ------------------------------ Quicksort

## 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.

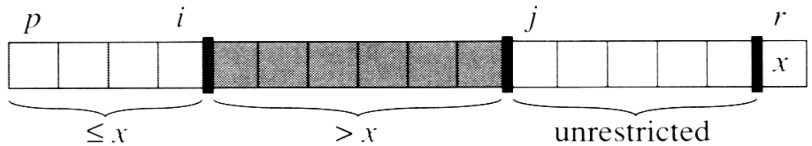
### *Quicksort(A, p, r)*

```
If p < r then
    q ← Partition(A, p, r)        /* divide */
    Quicksort(A, p, q-1)          /* conquer */
    Quicksort(A, q+1, r)          /* conquer */
```

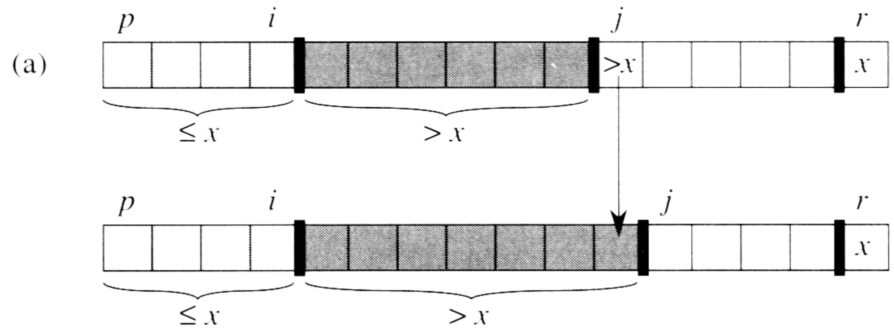
### *Partition(A, p, r)*

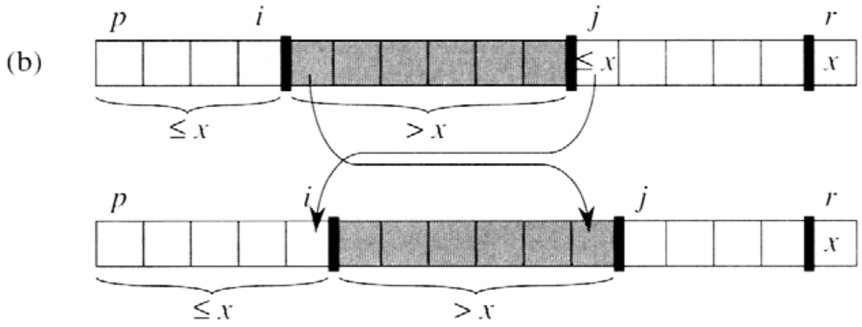
```
x ← A[r]
i ← p-1;
for j← p to r-1
   do if A[j] ≤ x
      then   i ← i+1
             exchange A[i]↔A[j]
exchange A[i+1]↔A[r]
return i+1
```

### Meaning of *i* and *j*:

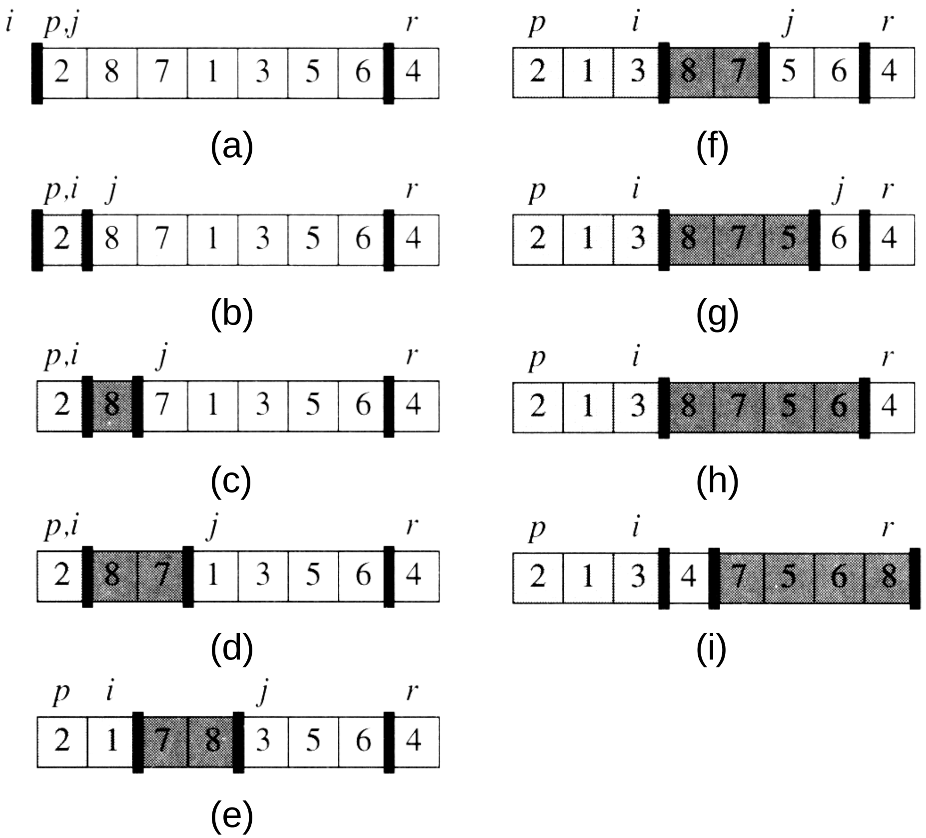



### How *i* and *j* changes:






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






## 7.4 Analysis

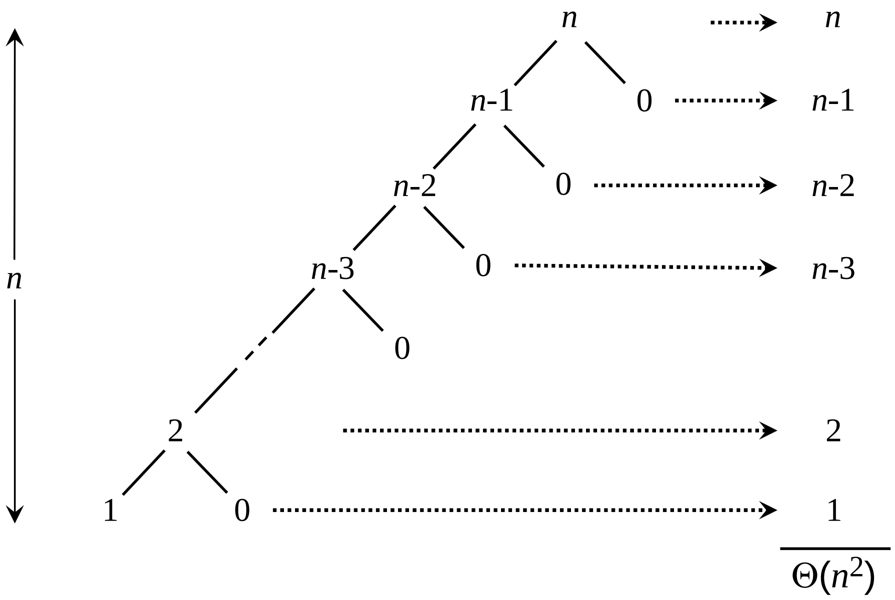
**Worst-case: $\Theta(n^2)$ (occurs for sorted input)**

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

Guessing: $T(n) \le c\, n^2 = O(n^2)$
Substituting:

$$T(n) \le \max_{0 \le k \le n-1} \{ck^2 + c(n-k-1)^2\} + \Theta(n)$$
$$\le c \max_{0 \le k \le n-1} \{k^2 + (n-k-1)^2\} + \Theta(n)$$
$$\le c(n\text{-}1)^2 + \Theta(n)$$
$$\le cn^2 - c(2n\text{-}1) + \Theta(n)$$
$$\le cn^2 \quad \text{(by picking } c \text{ large enough)}$$

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




**Average-case: $\Theta$(*n* lg *n*)**
(Assume that all elements are distinct.)

- $T(n)=O(n+X)$, 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, 7, 9}. Then, {3, 2} won't be compared with {7, 9} anymore.



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

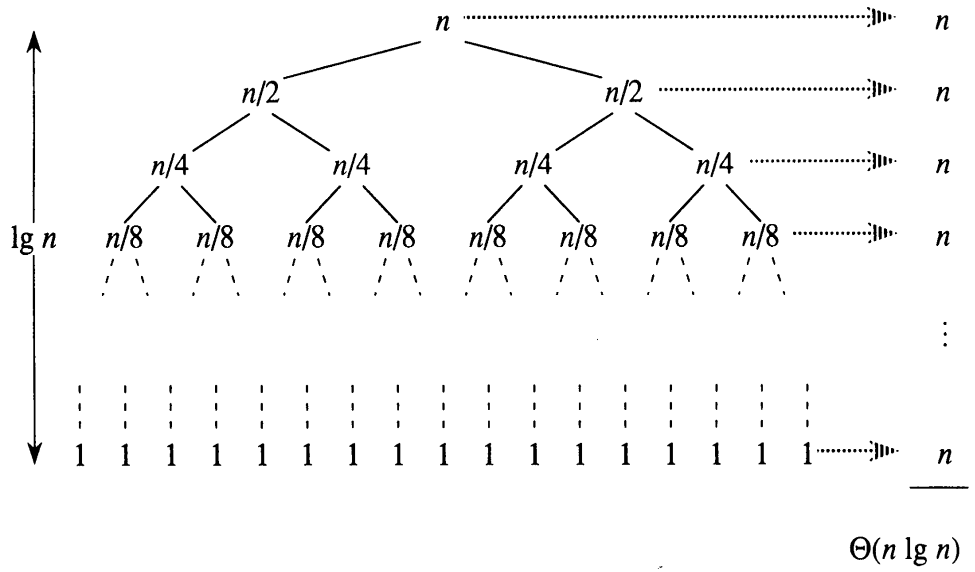
$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1}$$

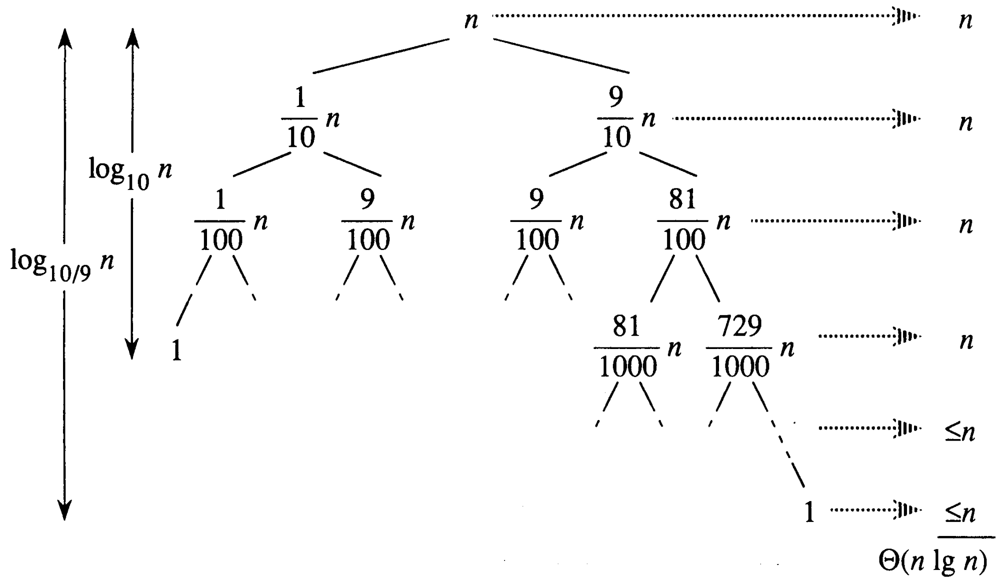
$$= \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1}$$

$$< \sum_{i=1}^{n-1} \sum_{k=1}^{n} \frac{2}{k} \qquad \text{(Using Harmonic Series)}$$

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

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






- **Another analysis**

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

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

For simplicity, assume

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

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

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

(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)

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

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

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

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

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

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

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

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



## 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*);
  exchange(*A*[*r*], *A*[*i*]);
  **return** Partition(*A*, *p*, *r*)

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