طراحی و تحلیل الگوریتم ها

دکتر امیر لکی زاده

استادیار گروه مهندسی کامپیوتر دانشگاه قم

مرتب سازی سریع

- مرتب سازی سریع **Quick Sort**

worst case: $\theta(n^2)$ best case, average case: $\theta(n \log n)$

```
QUICKSORT(A, p, r)
if p < r
    then q ← PARTITION(A, p, r)
         QUICKSORT(A, p, q − 1)
         QUICKSORT(A, q + 1, r)
```

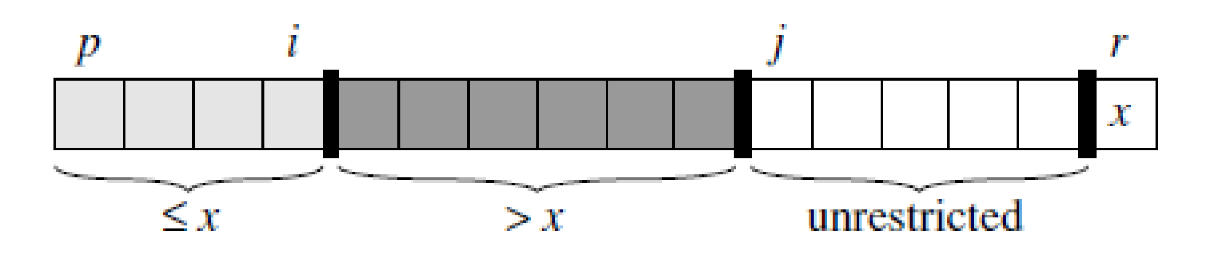
مراحل:

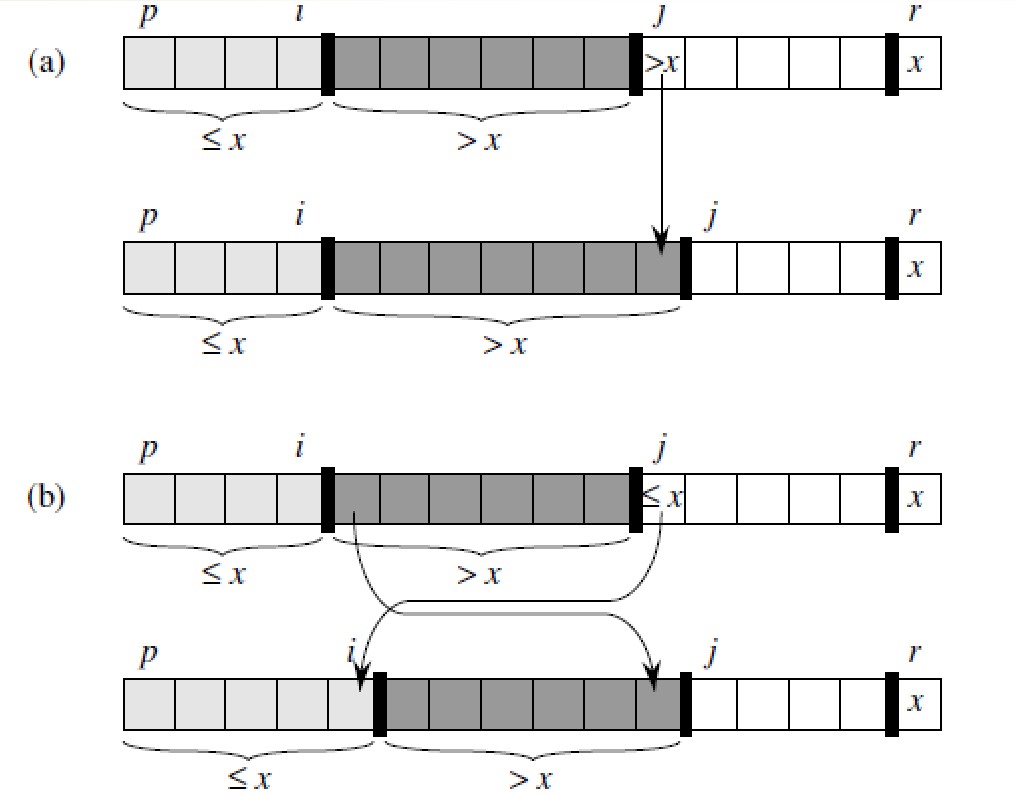
تقسیم: آرایه به دو زیر آرایه $A[P \dots q-1]$, $A[q+1 \dots r]$ تقسیم می شود.

حل: دو زیر آرایه با فراخوانی بازگشتی مرتب سازی سریع، مرتب می شوند.

ترکیب: چون دو زیر آرایه در جا مرتب می شوند نیاز به ترکیب آنها نیست.

مرتب سازی سریع

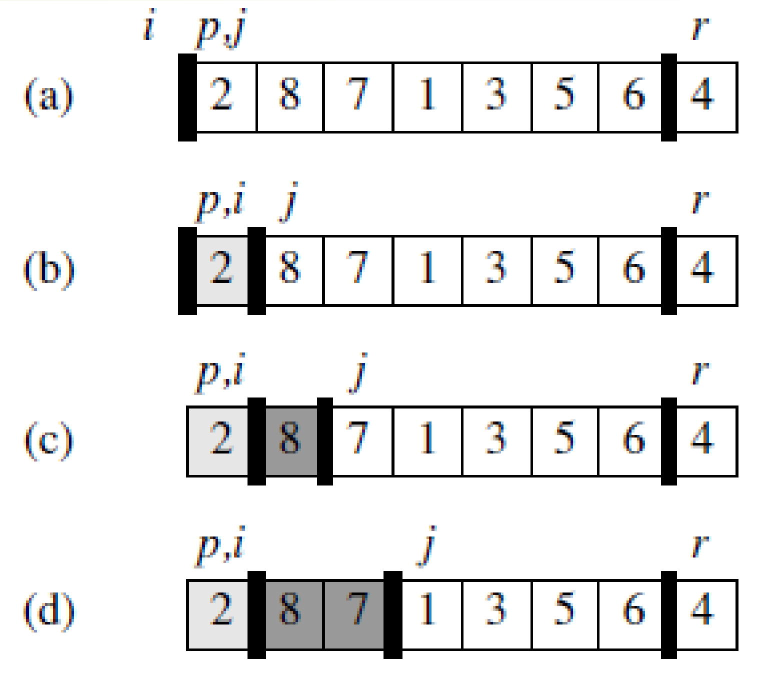


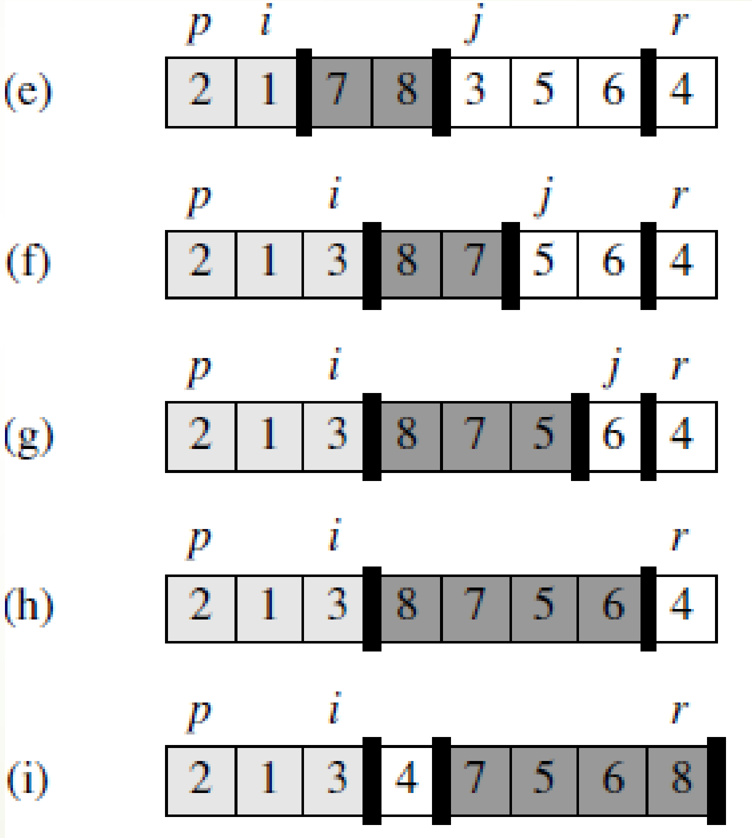


مرتب سازی سریع

```
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
```

مرتب سازی سریع





مرتب سازی سریع

worst case: $T(n) = T(0) + T(n-1) + \theta(n)$

بدترین حالت زمانی رخ می دهد که روال افراز، یک زیر مسئله با n-1 عنصر و زیر مسئله دیگر با صفر عنصر ایجاد کند.

$$T(n) = T(n-1) + Cn = T(n-2) + C(n-1) + Cn = \cdots$$
$$= T(1) + C(n + n - 1 + n - 2 + \cdots + 1) \rightarrow \ T(n) = \ O(n^2)$$

مرتب سازی سریع

- بهترین حالت، زمانی اتفاق می افتد که Pivot همواره در وسط آرایه قرار گیرد.

$$T(n) = 2T(n/2) + \theta(n)$$

$$\mathbf{T}(\boldsymbol{n}) = \boldsymbol{\Omega}(\mathbf{nlog\ n})$$

مرتب سازی سریع

4.2-5

Use a recursion tree to give an asymptotically tight solution to the recurrence $T(n) = T(\alpha n) + T((1-\alpha)n) + cn$, where α is a constant in the range $0 < \alpha < 1$ and $c > 0$ is also a constant.

مرتب سازی سریع

حالت میانگین Average case

۱. زمان اجرای Quick sort وابسته به PARTITION
۲. با هر فراخوانی PARTITION یک عنصر به عنوان Pivot در جای خود قرار می گیرد (n بار فراخوانی)

۳. در هر فراخوانی PARTITION تعدادی مقایسه انجام می شود + $\theta(1)$ عمل اضافی.
۴. X: تعداد کل مقایسه های انجام شده در n بار فراخوانی PARTITION

مرتب سازی سریع

$T(n) = O(X)$ $\quad E\left(T(n)\right) = O(?)$

Sorted: $< Z_i, Z_2, \ldots Z_n >$

Z_i: امین کوچکترین عنصر i

$Z_{ij}: \{Z_i, Z_{i+1}, \ldots Z_j\}$

$$X_{ij} = \begin{cases} 1 & Z_i \text{ is compare } Z_j \\ 0 & \text{else} \end{cases}$$

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

$$E\left(T(n)\right) = E(x) = E\left(\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}\right) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E(X_{ij})$$

$$E(x) = \sum_{x} P_r (X = x)x$$

$$E(x) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Pr\left(Z_i \text{ is compared } z_j\right) \times 1$$

مرتب سازی سریع

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} P_r(\, z_i \text{ or } z_j \;\text{ first Pivot choosen form } z_{ij}) \times 1$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} P_r(\, z_i \;\text{ first Pivot }) + P_r\,(Z_i \;\text{ is first Pivot})$$

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

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

$$\int_1^n \frac{1}{k} dk = \ln(n)$$

$$\le 2 \sum_{i=1}^{n-1} \ln(n) = C' \, n \log n$$

$$\sum \le \int$$

$$E\,(x) \le C' \, n \log \qquad\qquad E(x) = O\,(n \;\log n\,)$$

مرتب سازی سریع

```
RANDOMIZED-PARTITION(A, p, r)
i ← RANDOM(p, r)
exchange A[r] ↔ A[i]
return PARTITION(A, p, r)
```

The new quicksort calls RANDOMIZED-PARTITION in place of PARTITION:

```
RANDOMIZED-QUICKSORT(A, p, r)
if p < r
   then q ← RANDOMIZED-PARTITION(A, p, r)
        RANDOMIZED-QUICKSORT(A, p, q − 1)
        RANDOMIZED-QUICKSORT(A, q + 1, r)
```

مرتب سازی سریع

7-3 Stooge sort

Professors Howard, Fine, and Howard have proposed the following “elegant” sorting algorithm:

```
STOOGE-SORT(A, i, j)
if A[i] > A[j]
    then exchange A[i] ↔ A[j]
if i + 1 ≥ j
    then return
k ← ⌊(j − i + 1)/3⌋              ▷ Round down.
STOOGE-SORT(A, i, j − k)         ▷ First two-thirds.
STOOGE-SORT(A, i + k, j)         ▷ Last two-thirds.
STOOGE-SORT(A, i, j − k)         ▷ First two-thirds again.
```

a. Argue that, if $n = length[A]$, then STOOGE-SORT$(A, 1, length[A])$ correctly sorts the input array $A[1 \,.\,.\, n]$.

b. Give a recurrence for the worst-case running time of STOOGE-SORT and a tight asymptotic (Θ-notation) bound on the worst-case running time.