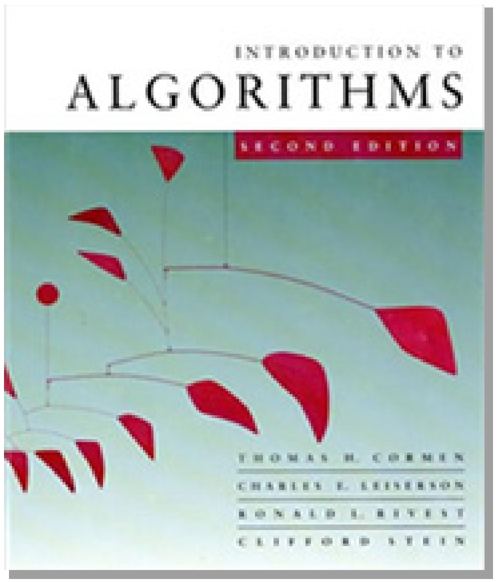
Algoritmalara Giriş

6.046J/18.401J

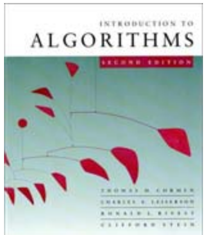


DERS 6

Sıra İstatistikleri

- Rastgele böl ve fethet
- Umulan sürenin çözümlemesi
- En kötü durum doğrusal süre sıra istatistikleri
- Çözümleme

Prof. Erik Demaine



Sıra istatistikleri

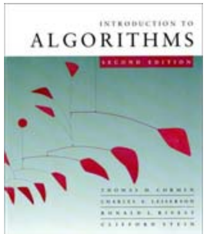
n elemanın i'ninci küçük değerini seçin (i ***rütbeli*** eleman).

- $i = 1$: ***minimum***; (en az)
- $i = n$: ***maximum***; (en çok)
- $i = \lfloor (n+1)/2 \rfloor$ veya $\lceil (n+1)/2 \rceil$: ***median***. (ortanca)

Saf algoritma: i'ninci elemanı sırala ve dizinle.

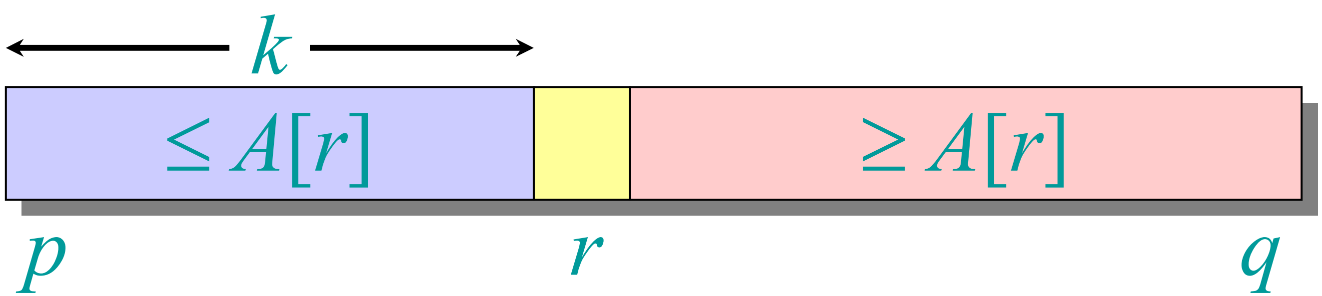
En kötü yürütüm süresi $= \Theta(n \lg n) + \Theta(1)$
$= \Theta(n \lg n)$,

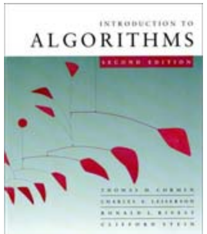
birleştirme veya yığin sıralaması kullan (*çabuk sır. değil*).



Rastgele böl-ve-fethet algoritması

RAND-SELECT(A, p, q, i) ▹ $A[p \,.\, q]$'nun $\boldsymbol{i}$'nci en küçüğü
  if $p = q$ **then return** $A[p]$
  $r \leftarrow$ RAND-PARTITION(A, p, q) (Rastgele bölüntü)
  $k \leftarrow r - p + 1$ ▹ $k = \text{rank}(A[r])$ (rütbeli)
  if $i = k$ **then return** $A[r]$
  if $i < k$
    then return RAND-SELECT$(A, p, r - 1, i)$
    else return RAND-SELECT$(A, r + 1, q, i - k)$





Örnek

$i = 7$'nci en küçük olarak seçin:

6	10	13	5	8	3	2	11

$i = 7$

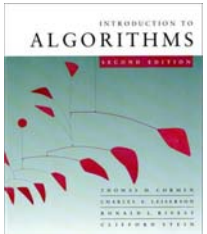
pivot ***(esas eleman)***

Partition (Bölüntü):

2	5	3	6	8	13	10	11

$k = 4$

$7 - 4 = 3$ 'üncü küçüğü özyinelemeyle seçin.



Çözümlemede sezgi(öngörü)

(Bugünkü çözümlemelerin hepsinde tüm elemanların farklı olduğu varsayılıyor.)

Şanslı:

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

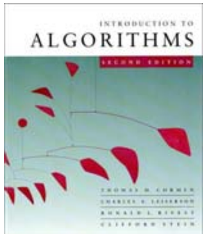
$n^{\log_{10/9} 1} = n^0 = 1$
DURUM 3

Şanssız:

$$T(n) = T(n-1) + \Theta(n) = \Theta(n^2)$$

aritmetik seri

Sıralamadan daha kötü!



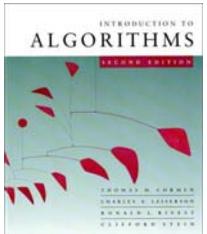
Umulan süre çözümlemesi

Çözümleme rastgele çabuk sıralamanın benzeri ama bazı farkları var.

$T(n)$ = Rastgele-seçim yürütüm süresinin rastgele değişkeni olsun (n boyutlu bir girişte), ve rastgele sayılar birbirinden bağımsız olsun.

$k = 0, 1, \ldots, n-1$ için ***göstergesel rastgele değişkeni*** tanımlayın.

$$X_k = \begin{cases} 1 & \text{eğer BÖLÜNTÜ } k : n-k-1 \text{ bölmeli ise,} \\ 0 & \text{diğer durumlarda.} \end{cases}$$



Çözümleme (devam)

Bir üst sımır elde etmek için, i' ninci elemanın her zaman bölüntünün büyük bölgesinde olduğunu varsayın:

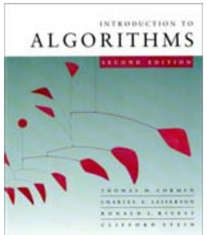
$$T(n) = \begin{cases} T(\max\{0, n-1\}) + \Theta(n), & 0 : n-1 \text{ bölünmesi,} \\ T(\max\{1, n-2\}) + \Theta(n), & 1 : n-2 \text{ bölünmesi,} \\ \vdots & \\ T(\max\{n-1, 0\}) + \Theta(n), & n-1 : 0 \text{ bölünmesi,} \end{cases}$$

$$= \sum_{k=0}^{n-1} X_k \left(T(\max\{k, n-k-1\}) + \Theta(n) \right).$$

Umulanın hesaplanması

$$E[T(n)] = E\left[\sum_{k=0}^{n-1} X_k \left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

Her iki taraftaki umaulanları bulun.

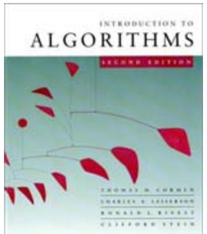


Umulanın hesaplanması

$$E[T(n)] = E\left[\sum_{k=0}^{n-1} X_k \left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

$$= \sum_{k=0}^{n-1} E\left[X_k \left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

Umulanın doğrusallığı.



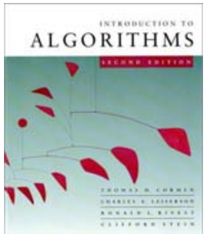
Umulanın hesaplanması

$$E[T(n)] = E\left[\sum_{k=0}^{n-1} X_k \left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

$$= \sum_{k=0}^{n-1} E\left[X_k \left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

$$= \sum_{k=0}^{n-1} E[X_k] \cdot E[T(\max\{k, n-k-1\}) + \Theta(n)]$$

X_k'nın diğer rastgele seçimlerden bağımsızlığı.



Umulanın hesaplanması

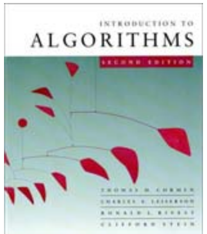
$$E[T(n)] = E\left[\sum_{k=0}^{n-1} X_k \left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

$$= \sum_{k=0}^{n-1} E\left[X_k \left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

$$= \sum_{k=0}^{n-1} E[X_k] \cdot E[T(\max\{k, n-k-1\}) + \Theta(n)]$$

$$= \frac{1}{n} \sum_{k=0}^{n-1} E[T(\max\{k, n-k-1\})] + \frac{1}{n} \sum_{k=0}^{n-1} \Theta(n)$$

Umulanın doğrusallığı; $E[X_k] = 1/n$.



Umulanın hesaplanması

$$E[T(n)] = E\left[\sum_{k=0}^{n-1} X_k \left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

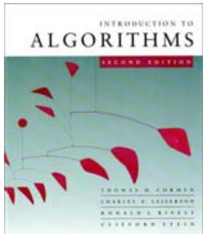
$$= \sum_{k=0}^{n-1} E\left[X_k \left(T(\max\{k, n-k-1\}) + \Theta(n)\right)\right]$$

$$= \sum_{k=0}^{n-1} E[X_k] \cdot E[T(\max\{k, n-k-1\}) + \Theta(n)]$$

$$= \frac{1}{n}\sum_{k=0}^{n-1} E[T(\max\{k, n-k-1\})] + \frac{1}{n}\sum_{k=0}^{n-1}\Theta(n)$$

$$\leq \frac{2}{n}\sum_{k=\lfloor n/2 \rfloor}^{n-1} E[T(k)] + \Theta(n)$$

Üstteki terimler iki kez görünüyor.



Karmaşık yineleme

(Ama çabuk sıralamanınki kadar karmaşık değil.)

$$E[T(n)] = \frac{2}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} E[T(k)] + \Theta(n)$$

Kanıtla: $E[T(n)] \le cn$ sabiti için $c > 0$.

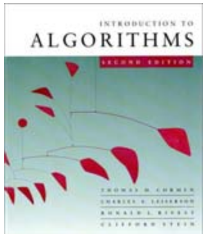
- c sabiti öyle büyük seçilebilir ki, $E[T(n)] \le cn$ tüm taban durumlarında geçerli olur .

Veri: $\sum_{k=\lfloor n/2 \rfloor}^{n-1} k \le \frac{3}{8} n^2$ (alıştırma).

Yerine koyma metodu

$$E[T(n)] \leq \frac{2}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} ck + \Theta(n)$$

Tümevarım hipotezini yerleştirin.

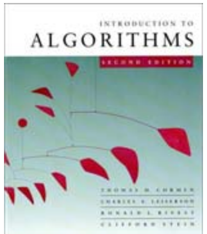


Substitution method

$$E[T(n)] \le \frac{2}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} ck + \Theta(n)$$

$$\le \frac{2c}{n}\left(\frac{3}{8}n^2\right) + \Theta(n)$$

Veriyi kullanın.



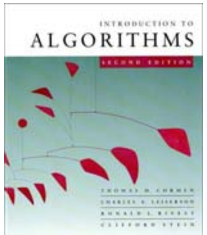
Yerine koyma metodu

$$E[T(n)] \le \frac{2}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} ck + \Theta(n)$$

$$\le \frac{2c}{n}\left(\frac{3}{8}n^2\right) + \Theta(n)$$

$$= cn - \left(\frac{cn}{4} - \Theta(n)\right)$$

istenen* – *kalan şeklinde gösterin.



Yerine koyma metodu

$$E[T(n)] \le \frac{2}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} ck + \Theta(n)$$

$$\le \frac{2c}{n}\left(\frac{3}{8}n^2\right) + \Theta(n)$$

$$= cn - \left(\frac{cn}{4} - \Theta(n)\right)$$

$$\le cn\,,$$

c yeterince büyük seçilirse
$cn/4$, $\Theta(n)$'nin üstünde olur.

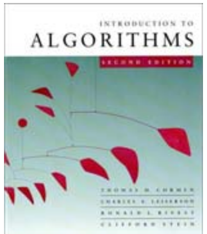
Rastgele sıra istatistik seçiminin özeti

- Hızlı çalışır: doğrusal umulan süre.
- Pratikte mükemmel bir algoritma.
- Ama, en kötü durumu ***-ok***kötü: $\Theta(n^2)$.

Q. En kötü durumda doğrusal zamanda çalışan bir algoritma var mıdır ?

A. Evet, Blum, Floyd, Pratt, Rivest ve Tarjan [1973] sayesinde vardır.

FİKİR: İyi bir pivotu yinelemeyle üretmek.



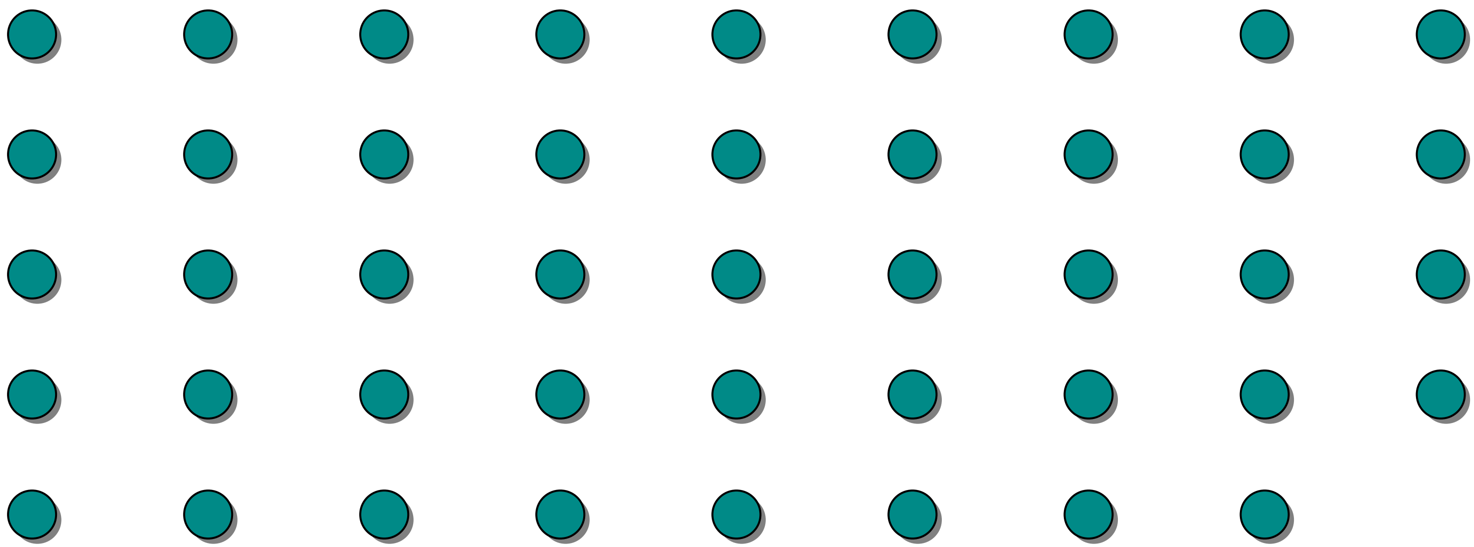
En kötü durum doğrusal-zaman sıra istatistikleri

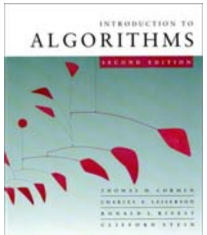
SELECT(i, n)

1. Divide the n elements into groups of 5. Find the median of each 5-element group by rote.
2. Recursively SELECT the median x of the $\lfloor n/5 \rfloor$ group medians to be the pivot.
3. Partition around the pivot x. Let $k = \text{rank}(x)$.
4. **if** $i = k$ **then return** x
   elseif $i < k$
   then recursively SELECT the ith smallest element in the lower part
   else recursively SELECT the $(i–k)$th smallest element in the upper part

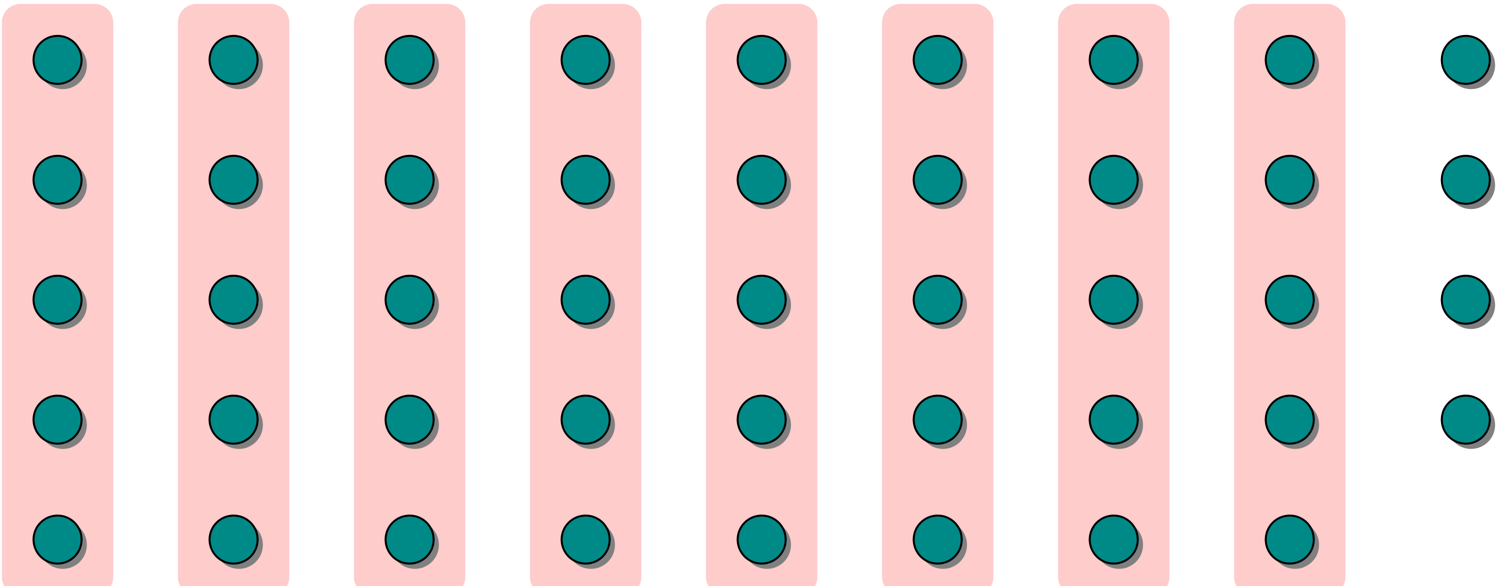
Same as RAND-SELECT (steps 3–4)

Choosing the pivot

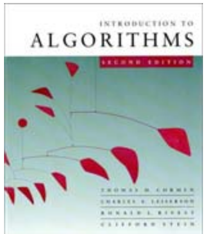




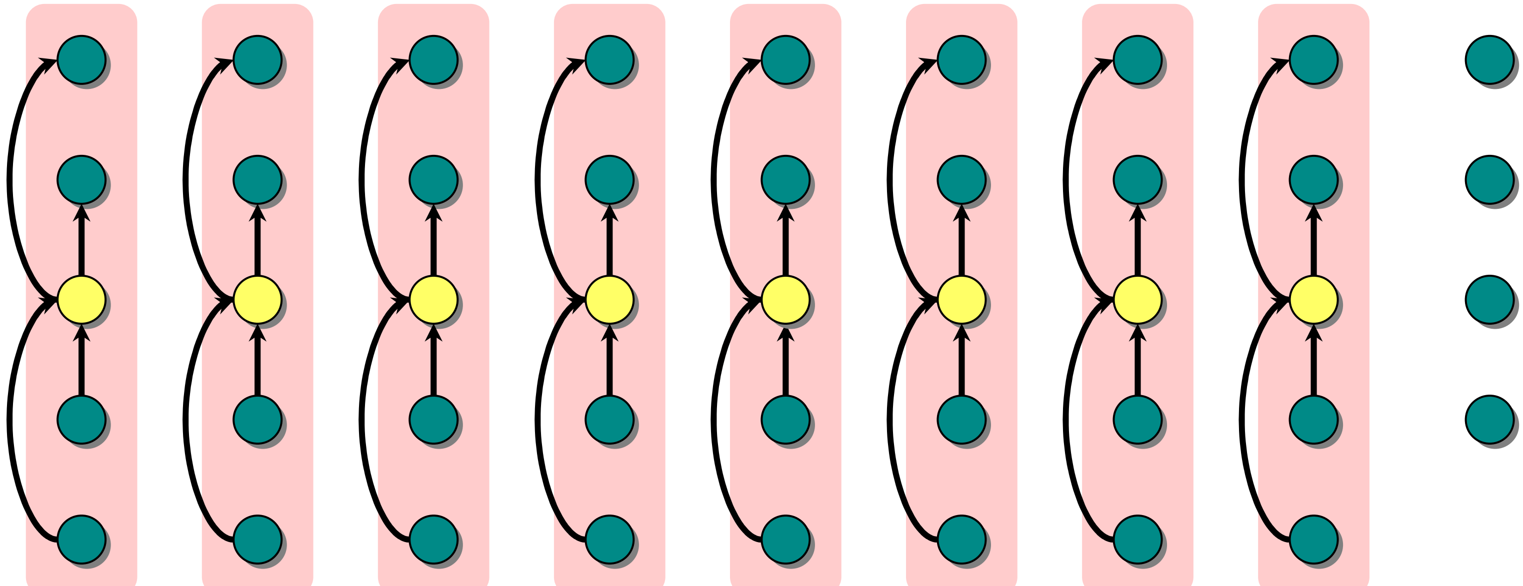
Choosing the pivot



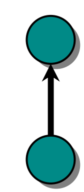
1. Divide the n elements into groups of 5.

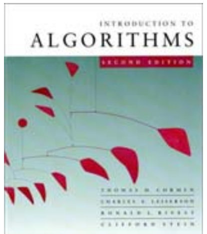


Choosing the pivot

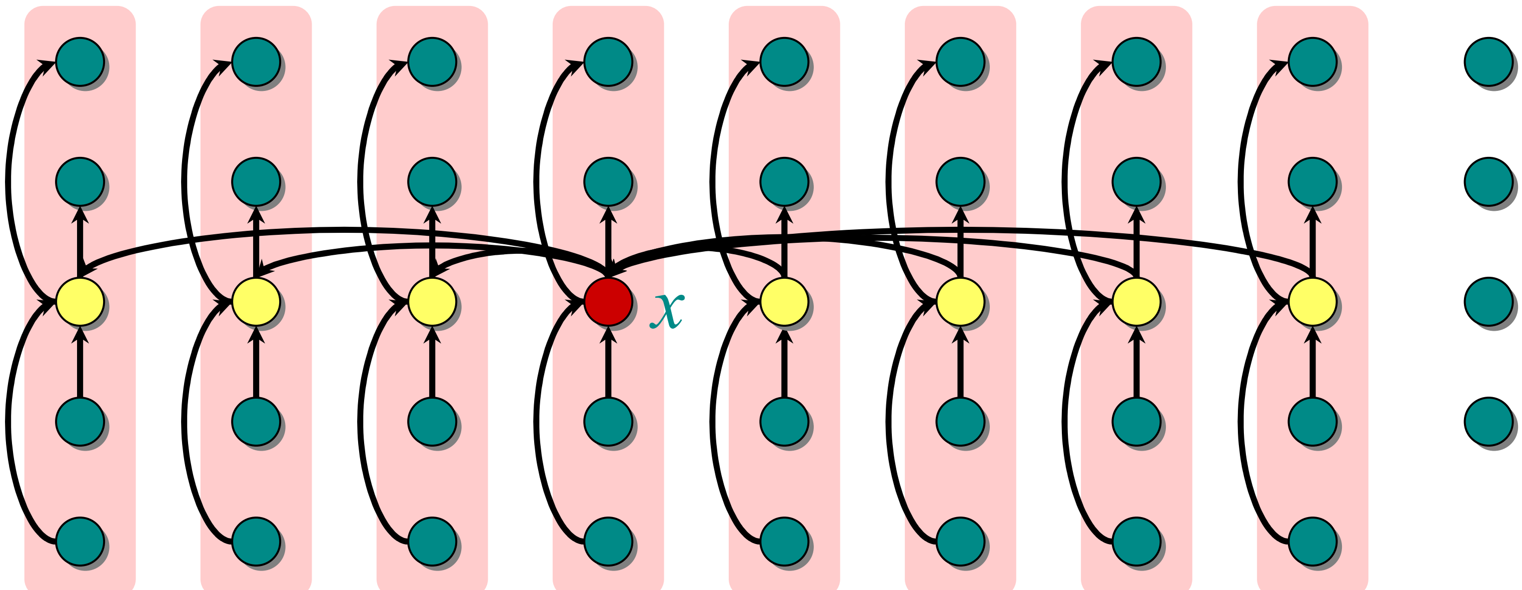


1. Divide the n elements into groups of 5. Find the median of each 5-element group by rote.

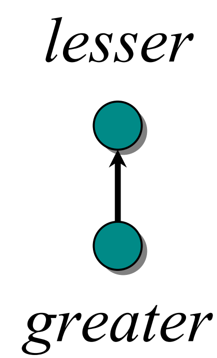


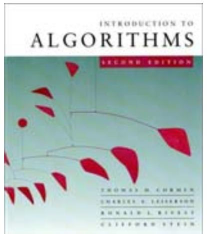


Choosing the pivot

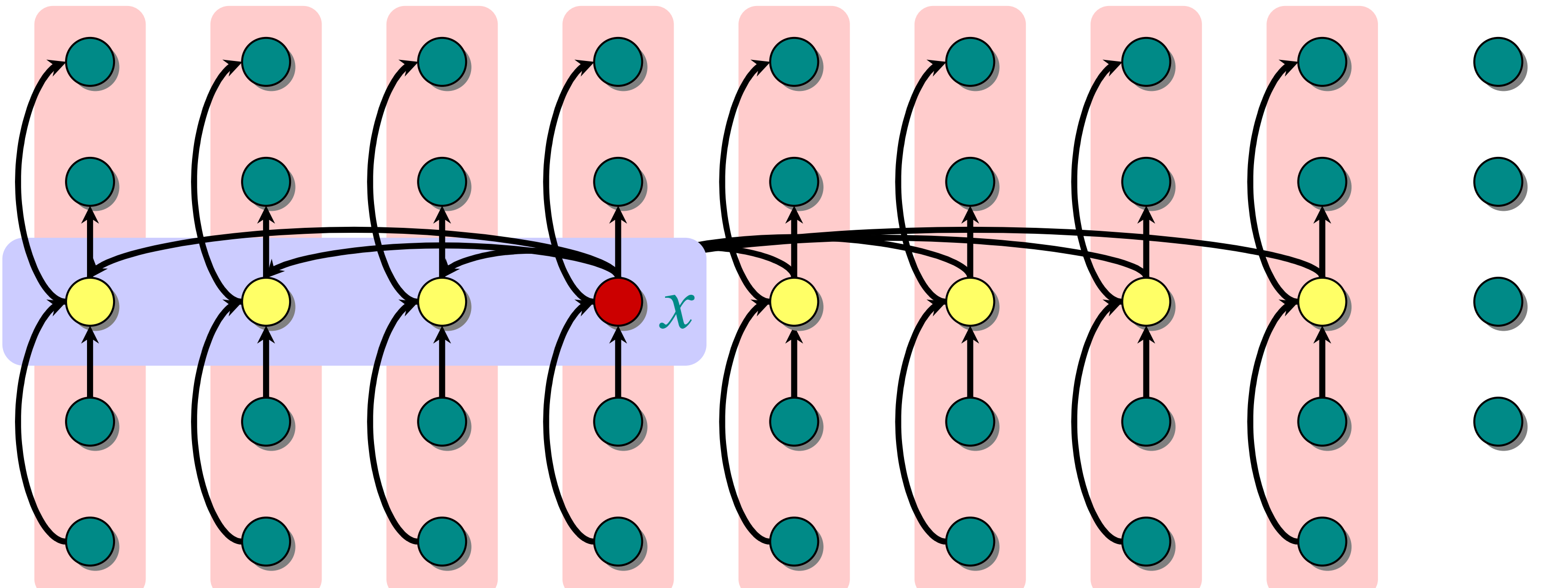


1. Divide the n elements into groups of 5. Find the median of each 5-element group by rote.
2. Recursively SELECT the median x of the $\lfloor n/5 \rfloor$ group medians to be the pivot.

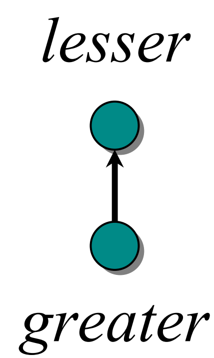


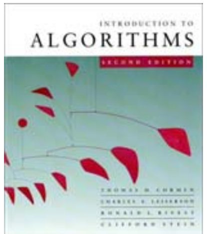


Analysis

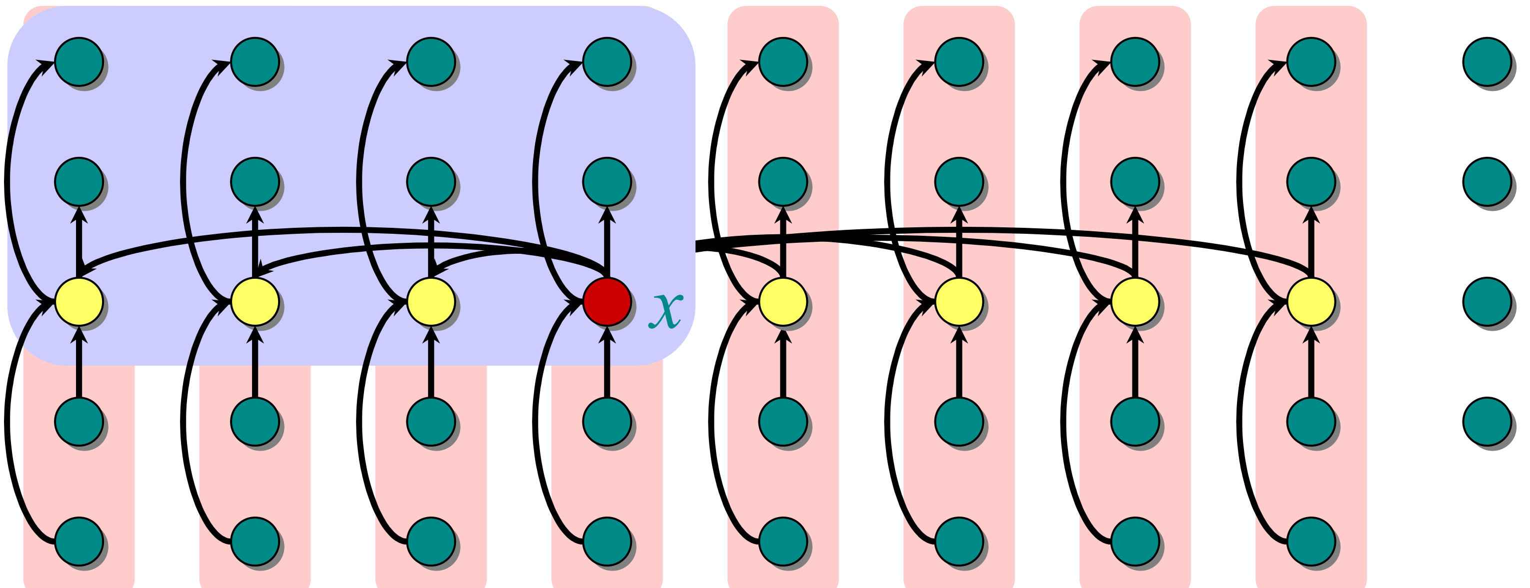


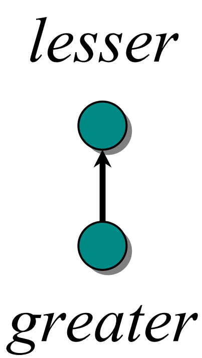
At least half the group medians are $\leq x$, which is at least $\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$ group medians.





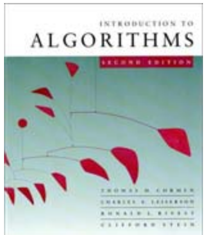
Analysis (Assume all elements are distinct.)



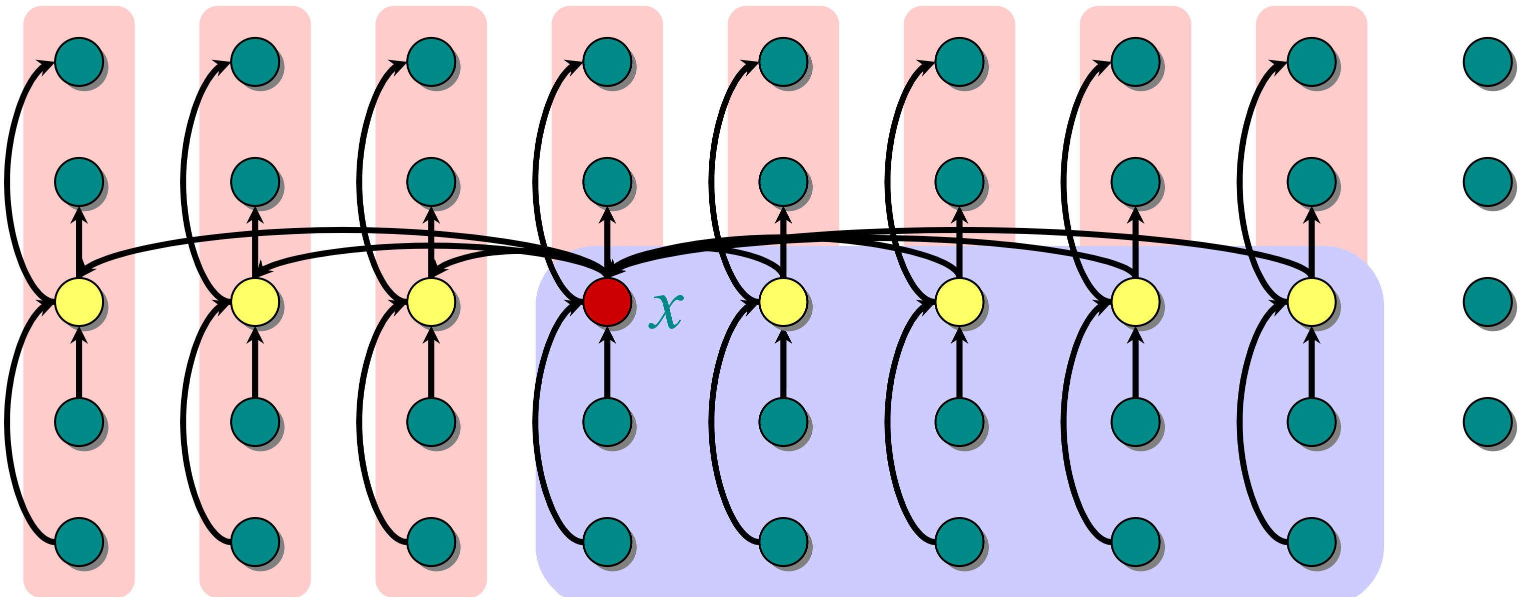


At least half the group medians are $\leq x$, which is at least $\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$ group medians.

- Therefore, at least $3 \lfloor n/10 \rfloor$ elements are $\leq x$.

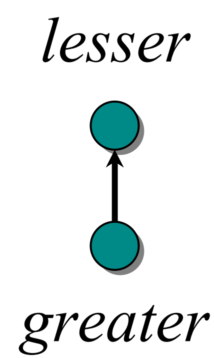


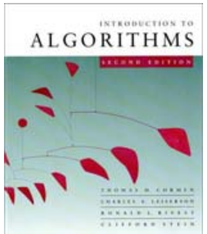
Analysis (Assume all elements are distinct.)



At least half the group medians are $\leq x$, which is at least $\lfloor \lfloor n/5 \rfloor /2 \rfloor = \lfloor n/10 \rfloor$ group medians.

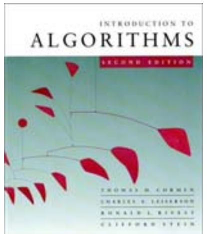
- Therefore, at least $3\lfloor n/10 \rfloor$ elements are $\leq x$.
- Similarly, at least $3\lfloor n/10 \rfloor$ elements are $\geq x$.





Minor simplification

- For $n \geq 50$, we have $3\lfloor n/10 \rfloor \geq n/4$.
- Therefore, for $n \geq 50$ the recursive call to SELECT in Step 4 is executed recursively on $\leq 3n/4$ elements.
- Thus, the recurrence for running time can assume that Step 4 takes time $T(3n/4)$ in the worst case.
- For $n < 50$, we know that the worst-case time is $T(n) = \Theta(1)$.



Developing the recurrence

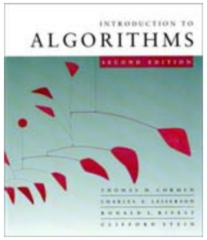
$T(n)$ — SELECT(i, n)

$\Theta(n)$ — 1. Divide the n elements into groups of 5. Find the median of each 5-element group by rote.

$T(n/5)$ — 2. Recursively SELECT the median x of the $\lfloor n/5 \rfloor$ group medians to be the pivot.

$\Theta(n)$ — 3. Partition around the pivot x. Let $k = \text{rank}(x)$.

$T(3n/4)$ — 4. **if** $i = k$ **then return** x
elseif $i < k$
then recursively SELECT the ith smallest element in the lower part
else recursively SELECT the $(i–k)$th smallest element in the upper part



Solving the recurrence

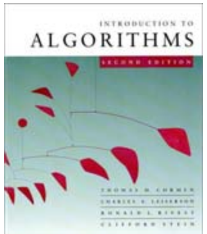
$$T(n) = T\left(\frac{1}{5}n\right) + T\left(\frac{3}{4}n\right) + \Theta(n)$$

Substitution:

$T(n) \le cn$

$$\begin{aligned} T(n) &\le \frac{1}{5}cn + \frac{3}{4}cn + \Theta(n) \\ &= \frac{19}{20}cn + \Theta(n) \\ &= cn - \left(\frac{1}{20}cn - \Theta(n)\right) \\ &\le cn \quad , \end{aligned}$$

if c is chosen large enough to handle both the $\Theta(n)$ and the initial conditions.



Conclusions

- Since the work at each level of recursion is a constant fraction (19/20) smaller, the work per level is a geometric series dominated by the linear work at the root.
- In practice, this algorithm runs slowly, because the constant in front of n is large.
- The randomized algorithm is far more practical.

Exercise: *Why not divide into groups of 3?*

