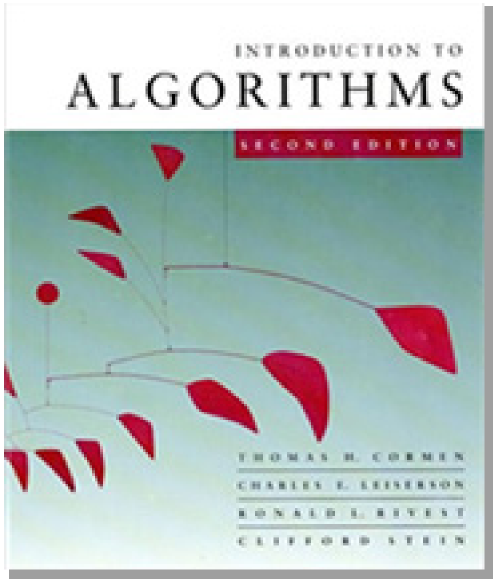
Algoritmalara Giriş

6.046J/18.401J



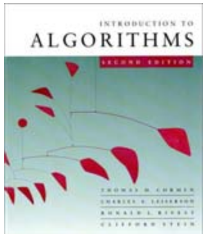
DERS 4

Çabuk sıralama

- Böl ve fethet
- Bölüntüler
- En kötü durum çözümlemesi
- Sezgi (Öngörü)
- Rastgele çabuk sıralama
- Çözümleme

Prof. Charles E. Leiserson

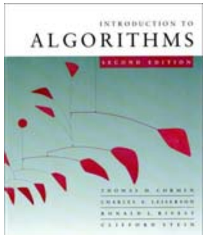
 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Çabuk sıralama

- C.A.R. Hoare tarafından 1962'de önerildi.
- Böl ve fethet algoritması.
- "Yerinde" sıralar (araya yerleştirme sıralamasında olduğu gibi; birleştirme sıralamasından farklı).
- (Ayar yapılırsa) çok pratik.

 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Böl ve fethet

n-elemanlı bir dizilimin çabuk sıralanması:

1. Böl: Dizilimi ***pivot (eksen sabit)*** x'in etrafında iki altdizilime bölüntüle; burada soldaki altdizilim elemanları $\leq x \leq$ sağdaki altdizilim elemanları olsun.

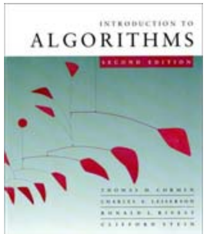
$\leq x$	x	$\geq x$

2. Fethet: İki altdizilimi özyinelemeli sırala.

3. Birleştir: Önemsiz.

Anahtar: *Doğrusal-zamanlı bölüntü altyordamı.*

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Bölüntüleme altyordamı

PARTITION (Bölüntü)(A, p, q) $\rhd$ $A[p \,..\, q]$

$x \leftarrow A[p]$ $\rhd$ pivot $= A[p]$

$i \leftarrow p$ *(eksen sabit)*

for $j \leftarrow p + 1$ **to** q

do if $A[j] \le x$ (öyleyse yap)

then $i \leftarrow i + 1$

exchange $A[i] \leftrightarrow A[j]$ (değiştir)

exchange $A[p] \leftrightarrow A[i]$ (değiştir)

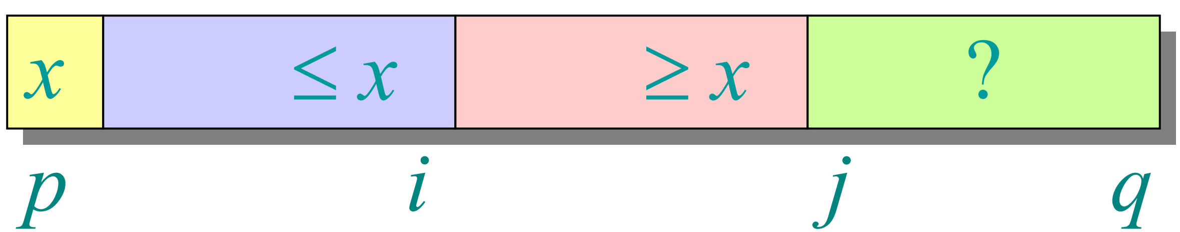
return i (dön)

Koşma süresi

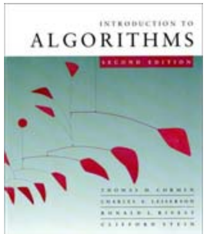
$= O(n)$

n eleman için

Değişmez:



 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



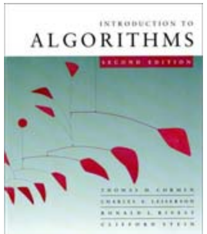
Bölüntüleme örneği

6	10	13	5	8	3	2	11
i	j						

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

Bölüntüleme örneği

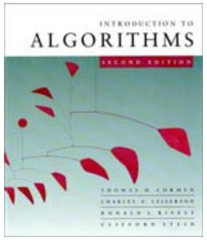
6	10	13	5	8	3	2	11
i	$\longrightarrow$	j					



Bölüntüleme örneği

6	10	13	5	8	3	2	11

i $\longrightarrow$ j



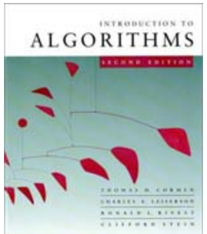
Bölüntüleme örneği

6	10	13	5	8	3	2	11

6	5	13	10	8	3	2	11

$\longrightarrow i \qquad\qquad j$

 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

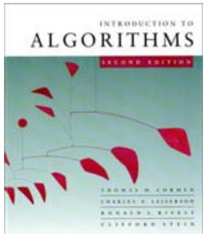


Bölüntüleme örneği

6	10	13	5	8	3	2	11

6	5	13	10	8	3	2	11
	i			j			

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

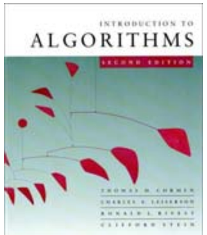


Bölüntüleme örneği

6	10	13	5	8	3	2	11

6	5	13	10	8	3	2	11
	i				j		

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Bölüntüleme örneği

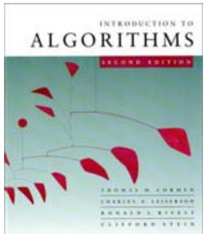
6	10	13	5	8	3	2	11

6	5	13	10	8	3	2	11

6	5	3	10	8	13	2	11

i j

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



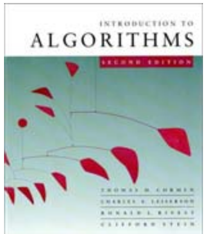
Bölüntüleme örneği

6	10	13	5	8	3	2	11

6	5	13	10	8	3	2	11

6	5	3	10	8	13	2	11
		i				*j*	

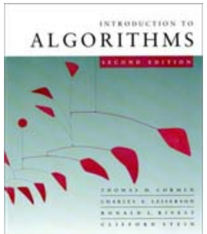
 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Bölüntüleme örneği

6	10	13	5	8	3	2	11
6	5	13	10	8	3	2	11
6	5	3	10	8	13	2	11
6	5	3	2	8	13	10	11

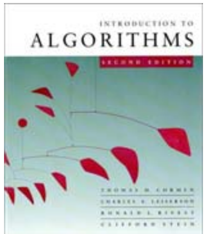
$\longrightarrow i$ $\qquad\qquad j$



Bölüntüleme örneği

6	10	13	5	8	3	2	11
6	5	13	10	8	3	2	11
6	5	3	10	8	13	2	11
6	5	3	2	8	13	10	11
			i				j

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Bölüntüleme örneği

6	10	13	5	8	3	2	11

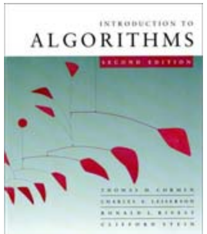
6	5	13	10	8	3	2	11

6	5	3	10	8	13	2	11

6	5	3	2	8	13	10	11

i $\longrightarrow j$

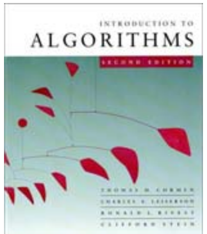
Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Bölüntüleme örneği

6	10	13	5	8	3	2	11
6	5	13	10	8	3	2	11
6	5	3	10	8	13	2	11
6	5	3	2	8	13	10	11
2	5	3	6	8	13	10	11
			i				

 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

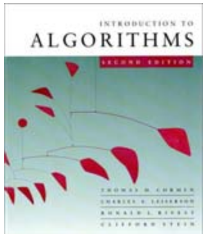


Çabuk sıralama (quicksort) için sözdekod

QUICKSORT(A, p, r)
    if $p < r$
        then $q \leftarrow$ PARTITION(A, p, r)
            QUICKSORT(A, p, q–1)
            QUICKSORT(A, q+1, r)

İlk arama: QUICKSORT(A, 1, n)

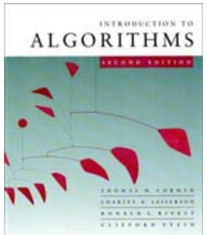
Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Çabuk sıralamanın çözümlemesi

- Tüm girdi elemanlarının farklı olduğunu varsayın.
- Pratikte, tekrarlayan girdi elemanları varsa, daha iyi algoritmalar vardır.

- n elemanı olan bir dizilimde $T(n) =$ en kötü koşma süresi olsun.

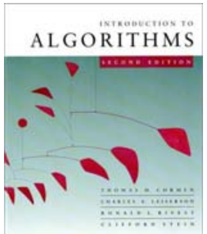


Çabuk sıralamanın en kötü durumu

- Girdiler sıralı ya da ters sıralı.
- En küçük yada en büyük elemanların etrafında bölüntüleme.
- Bölüntünün bir yanında hiç eleman yok.

$$
\begin{aligned}
T(n) &= T(0) + T(n-1) + \Theta(n) \\
&= \Theta(1) + T(n-1) + \Theta(n) \\
&= T(n-1) + \Theta(n) \\
&= \Theta(n^2) \qquad \textbf{\textit{(aritmetik seri)}}
\end{aligned}
$$

 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



En kötü durum özyineleme ağacı

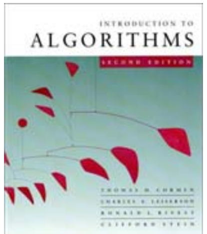
$$T(n) = T(0) + T(n-1) + cn$$

En kötü durum özyineleme ağacı

$$T(n) = T(0) + T(n-1) + cn$$

$T(n)$

 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

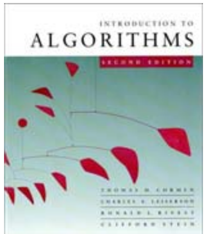


En kötü durum özyineleme ağacı

$$T(n) = T(0) + T(n-1) + cn$$

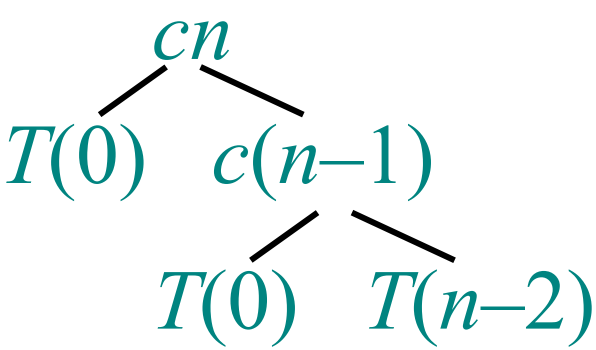
$$cn$$

$$T(0) \qquad T(n-1)$$

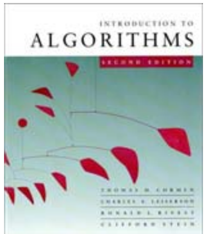


En kötü durum özyineleme ağacı

$$T(n) = T(0) + T(n–1) + cn$$

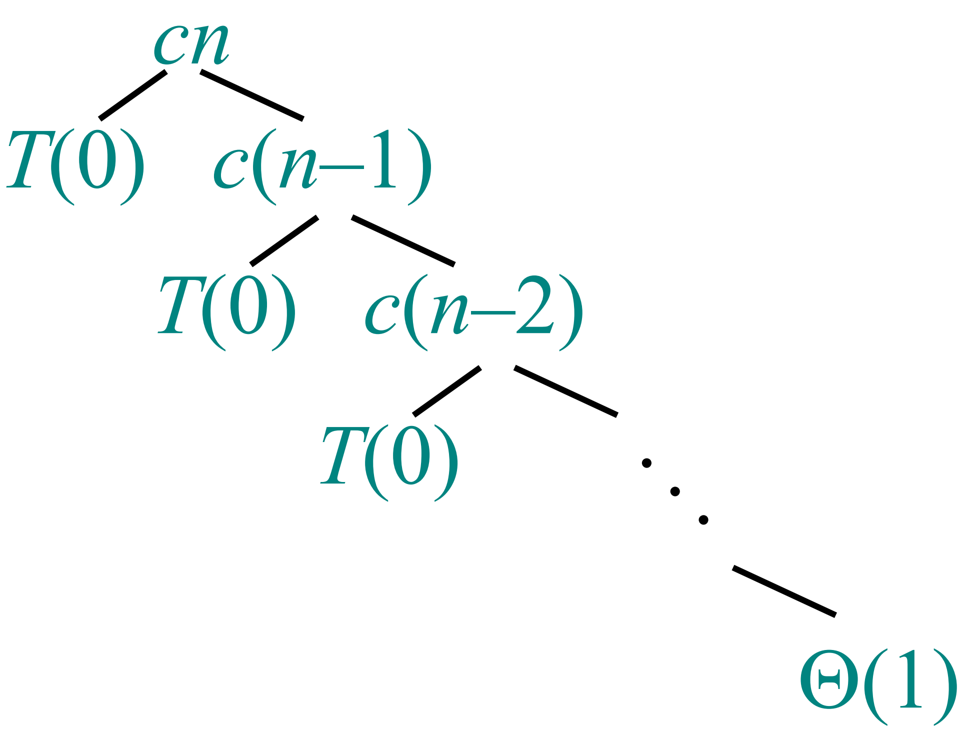


 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

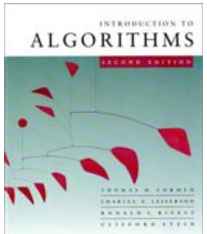


En kötü durum özyineleme ağacı

$$T(n) = T(0) + T(n-1) + cn$$

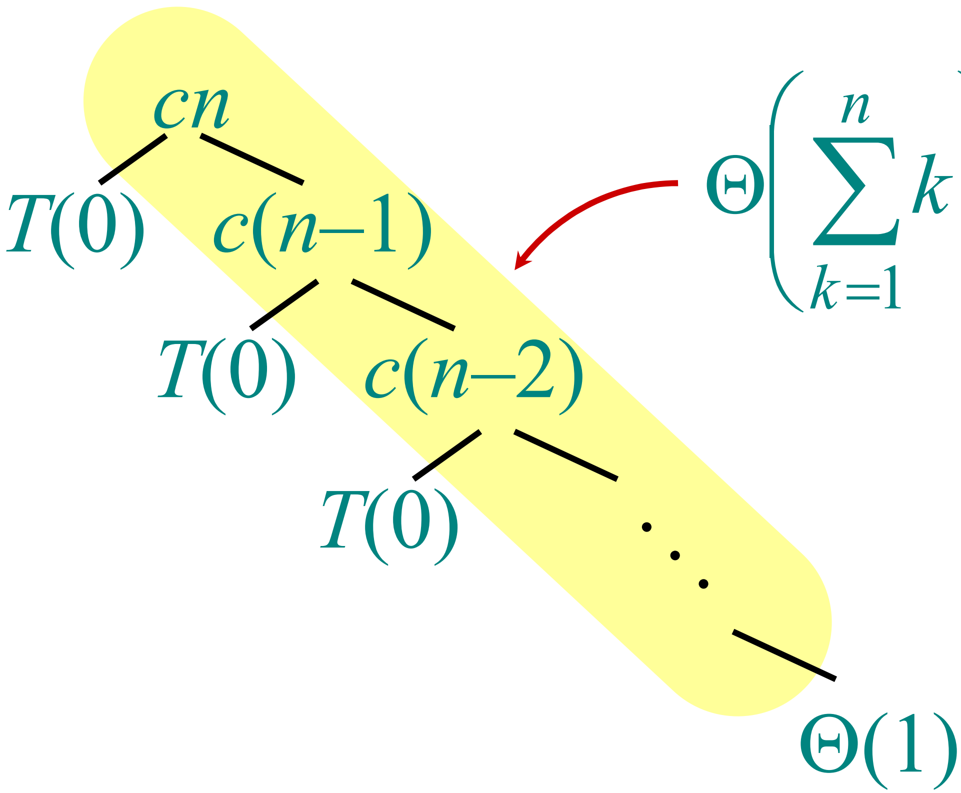


 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

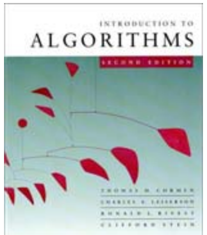


En kötü durum özyineleme ağacı

$$T(n) = T(0) + T(n-1) + cn$$

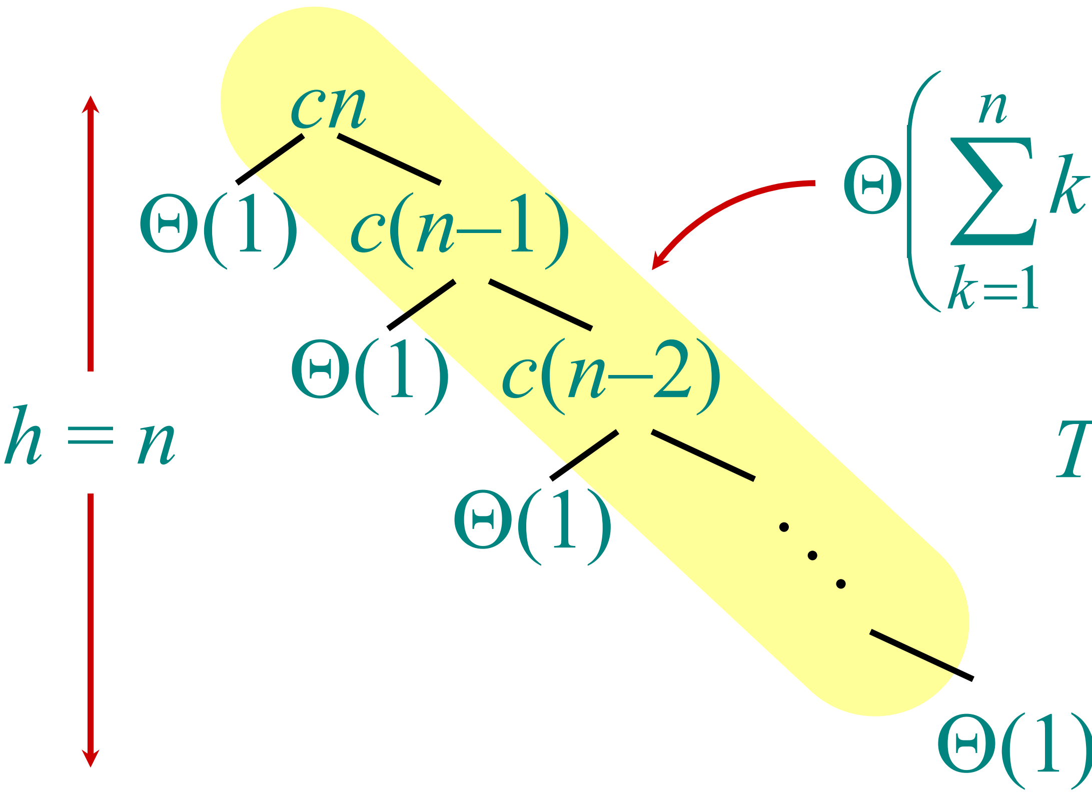


$$\Theta\left(\sum_{k=1}^{n} k\right) = \Theta\left(n^2\right)$$



En kötü durum özyineleme ağacı

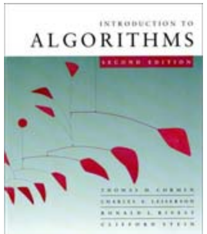
$$T(n) = T(0) + T(n-1) + cn$$



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

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

 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



En iyi durum çözümlemesi

(Yalnızca sezgi gelişimi amaçlı!)

Eğer şanslıysak, BÖLÜNTÜ dizilimi eşit böler:

$$\begin{aligned} T(n) &= 2T(n/2) + \Theta(n) \\ &= \Theta(n \lg n) \end{aligned}$$

(birleştirme sıralamasındaki gibi)

Ya bölünme her zaman $\frac{1}{10} : \frac{9}{10}$ oranındaysa?

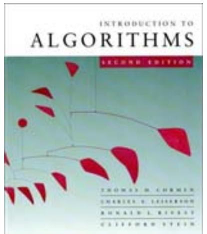
$$T(n) = T\left(\tfrac{1}{10}n\right) + T\left(\tfrac{9}{10}n\right) + \Theta(n)$$

Bu yinelemenin çözümü nedir?

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

“En iyiye yakın”durumun çözümlemesi

$$T(n)$$

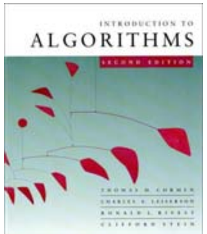


"En iyiye yakın"durumun çözümlemesi

$$cn$$

$$T\left(\tfrac{1}{10}n\right) \qquad\qquad T\left(\tfrac{9}{10}n\right)$$

 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

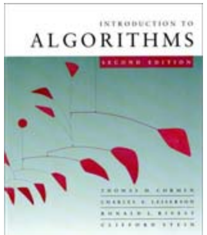


“En iyiye yakın”durumun çözümlemesi

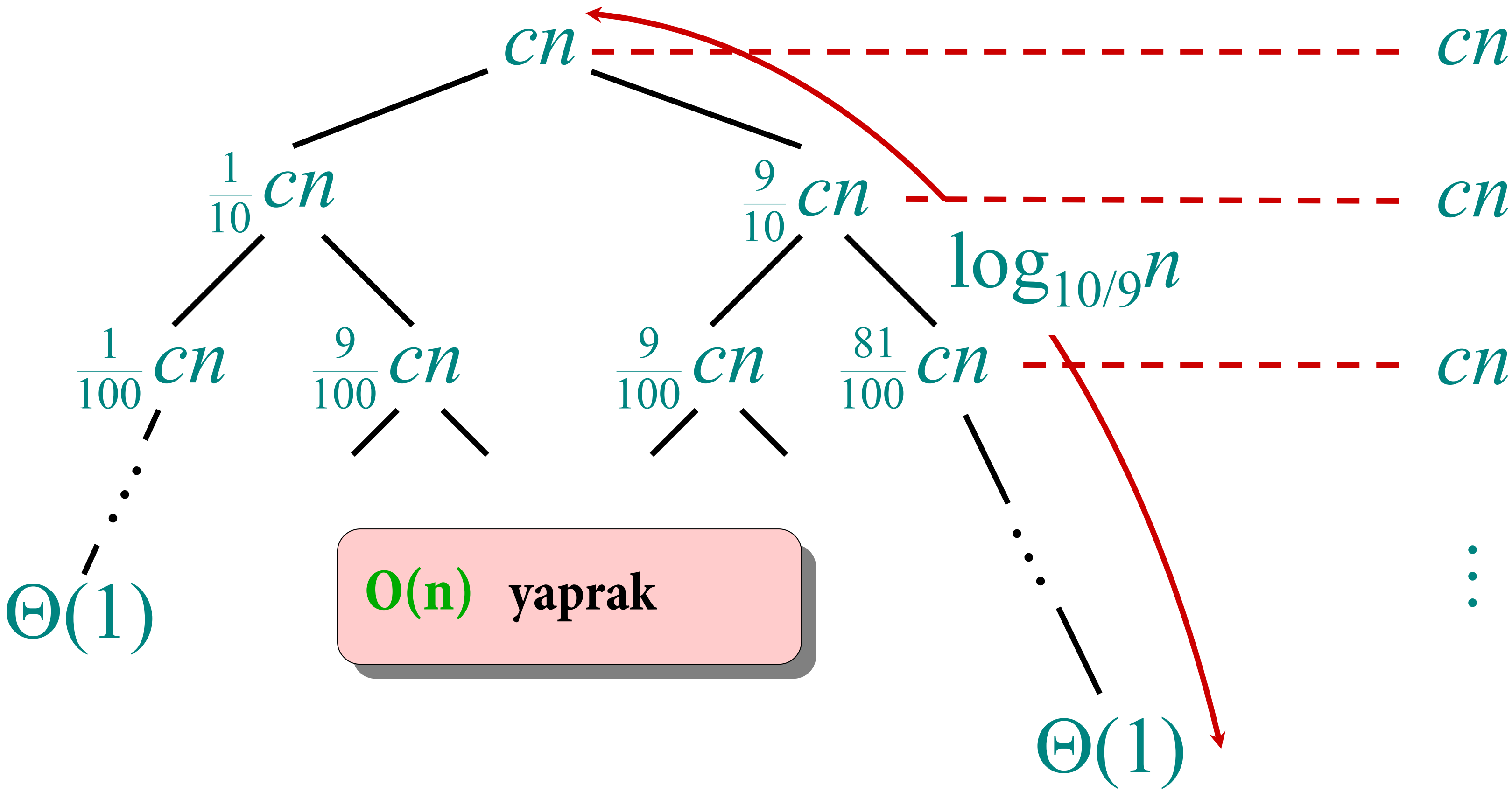
$$cn$$

$$\tfrac{1}{10}cn \qquad\qquad \tfrac{9}{10}cn$$

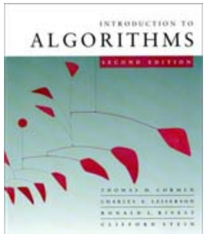
$$T\left(\tfrac{1}{100}n\right)\;T\left(\tfrac{9}{100}n\right)\qquad T\left(\tfrac{9}{100}n\right)\;T\left(\tfrac{81}{100}n\right)$$



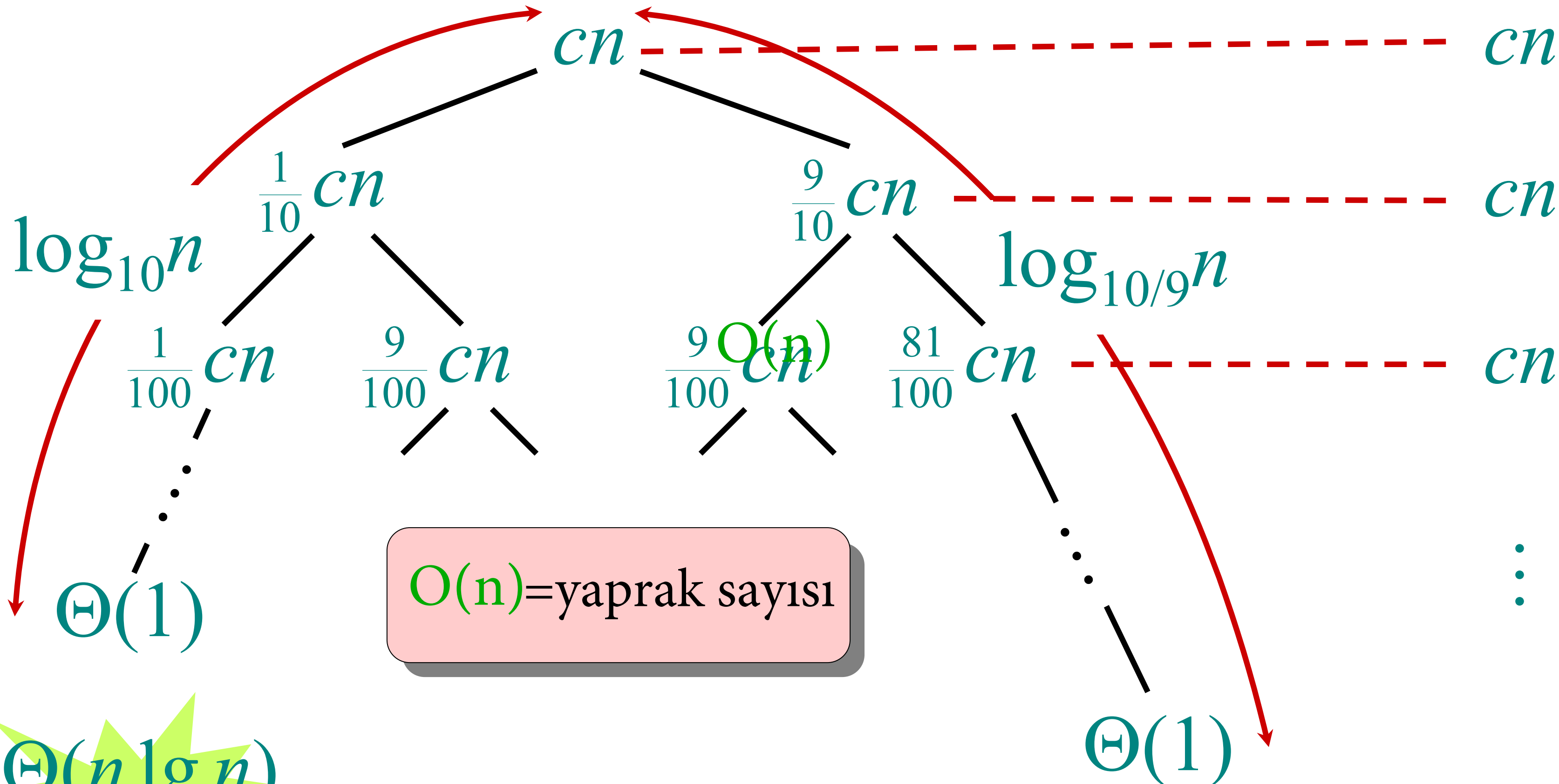
“En iyiye yakın”durumun çözümlemesi



Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



"En iyiye yakın"durumun çözümlemesi

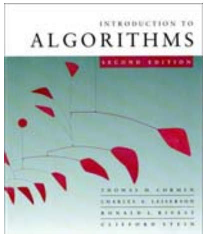


$\Theta(n \lg n)$

Şanslıyız!

$$cn \log_{10} n \le T(n) \le cn \log_{10/9} n + O(n)$$

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Daha fazla sezgi

Şanslı ve şanssız durumlar arasında sırayla gidip geldiğimizi varsayalım …

$$L(n) = 2U(n/2) + \Theta(n) \quad \textbf{\textit{şanslı durum}}$$

$$U(n) = L(n-1) + \Theta(n) \quad \textbf{\textit{şanssız durum}}$$

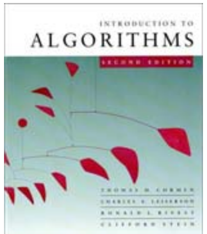
Çözelim:

$$\begin{aligned} L(n) &= 2(L(n/2-1) + \Theta(n/2)) + \Theta(n) \\ &= 2L(n/2-1) + \Theta(n) \\ &= \Theta(n \lg n) \end{aligned}$$

Şanslı!

Genellikle şanslı olmayı nasıl garanti ederiz?

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



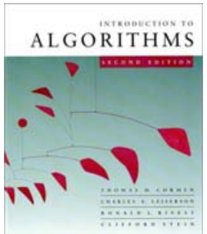
Rastgele çabuk sıralama

FİKİR: *Rastgele* bir eleman çevresinde bölüntü yap.

- Koşma süresi girişteki düzenden bağımsızdır.

- Girdideki dağılım konusunda herhangi bir varsayıma gerek yoktur.
- Hiçbir girdi en kötü durum davranışına neden olmaz.
- En kötü durum yalnızca rasgele sayı üretecinin çıkışına bağlıdır.

 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



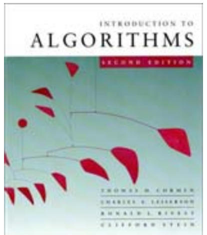
Rastgele çabuk sıralama çözümlemesi

n boyutlu ve sayıların bağımsız varsayıldığı bir girdinin, rastgele çabuk çözümlemesi için $T(n)$ = koşma süresinin rastgele değişkeni olsun.

$k = 0, 1, \ldots, n–1$, için ***indicator random variable (göstergesel rastgele değişken)***'i tanımlayın

$$X_k = \begin{cases} 1 & \text{eğer BÖLÜNTÜ bir } k : n–k–1 \text{ bölünme yaratıyorsa,} \\ 0 & \text{diğer durumlarda.} \end{cases}$$

$E[X_k] = \Pr\{X_k = 1\} = 1/n$, elemanların farklı olduğu varsayılırsa, her bölünme işleminin olasılığı aynıdır.

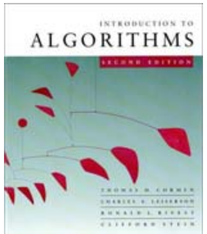


Çözümleme (devam)

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

$$= \sum_{k=0}^{n-1} X_k \big(T(k) + T(n-k-1) + \Theta(n) \big)$$

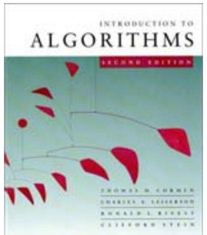
 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Beklenenin hesaplanması

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

Bekleneni her iki tarafta alın.

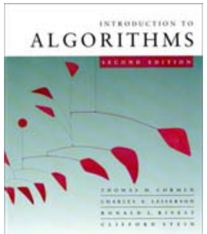


Beklenenin hesaplanması

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

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

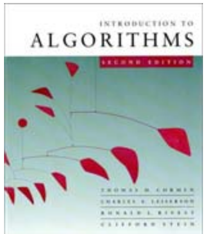
Beklenenin doğrusallığı.



Beklenenin hesaplanması

$$
\begin{aligned}
E[T(n)] &= E\left[\sum_{k=0}^{n-1} X_k\big(T(k)+T(n-k-1)+\Theta(n)\big)\right] \\
&= \sum_{k=0}^{n-1} E\big[X_k\big(T(k)+T(n-k-1)+\Theta(n)\big)\big] \\
&= \sum_{k=0}^{n-1} E[X_k]\cdot E[T(k)+T(n-k-1)+\Theta(n)]
\end{aligned}
$$

X_k 'nın diğer değişken seçeneklerden bağımsızlığı.



Beklenenin hesaplanması

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

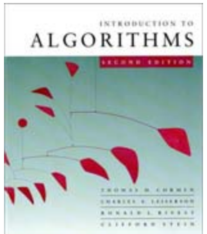
$$= \sum_{k=0}^{n-1} E\bigl[X_k\bigl(T(k) + T(n-k-1) + \Theta(n)\bigr)\bigr]$$

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

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

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

 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Beklenenin hesaplanması

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

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

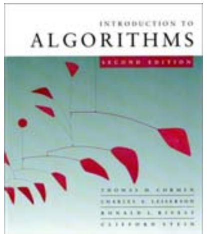
$$= \sum_{k=0}^{n-1} E[X_k] \cdot E[T(k) + T(n-k-1) + \Theta(n)]$$

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

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

Toplamlarda benzer terimler var.

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Karmaşık yineleme

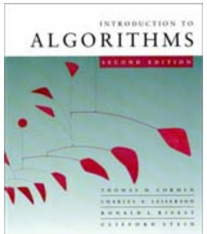
$$E[T(n)] = \frac{2}{n}\sum_{k=2}^{n-1} E[T(k)] + \Theta(n)$$

($k = 0, 1$ terimleri $\Theta(n)$ içine yedirilebilir.)

Kanıtla: $E[T(n)] \le a n \lg n$ ($a > 0$ sabiti için)

- a'yı öyle büyük seçin ki, yeterince küçük $n \ge 2$ için $a n \lg n$, $E[T(n)]$ 'ye göre başat olsun.

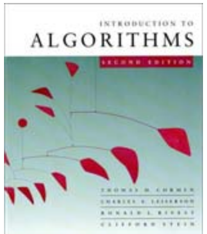
Kullan: $\displaystyle\sum_{k=2}^{n-1} k \lg k \le \tfrac{1}{2} n^2 \lg n - \tfrac{1}{8} n^2$ (egzersiz).



Yerine koyma metodu

$$E[T(n)] \le \frac{2}{n} \sum_{k=2}^{n-1} ak \lg k + \Theta(n)$$

Tümevarım hipotezini yerine koyun.



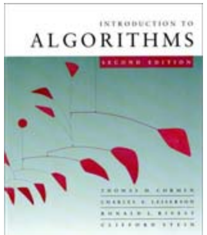
Yerine koyma metodu

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

$$\leq \frac{2a}{n} \left(\frac{1}{2} n^2 \lg n - \frac{1}{8} n^2 \right) + \Theta(n)$$

Bilineni kullanın.

 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Yerine koyma metodu

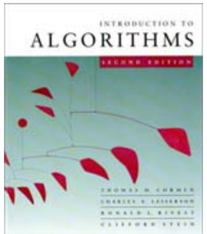
$$E[T(n)] \le \frac{2}{n}\sum_{k=2}^{n-1} ak\lg k + \Theta(n)$$

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

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

İstenen ***(desired)*** – kalan ***(residual)*** olarak ifade edin.

 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Yerine koyma metodu

$$E[T(n)] \le \frac{2}{n}\sum_{k=2}^{n-1} ak \lg k + \Theta(n)$$

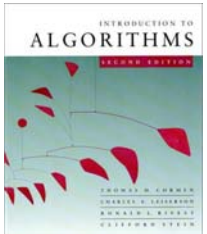
$$= \frac{2a}{n}\left(\frac{1}{2}n^2 \lg n - \frac{1}{8}n^2\right) + \Theta(n)$$

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

$$\le an \lg n\,,$$

eğer a yeterince büyük seçilir ve $an/4$ $\Theta(n)$'e göre başat olursa.

 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson



Pratikte çabuk sıralama

- Çabuk sıralama önemli bir genel maksatlı sıralama algoritmasıdır.
- Çabuk sıralama tipik olarak birleştirme sıralamasından iki kat daha hızlıdır.
- Çabuk sıralama ***kod ayarı*** uygulamasından önemli oranlarda yararlanır.
- Çabuk sıralama önbellekleme ve sanal bellek uygulamalarında oldukça uyumludur.

 Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson