# 4.Hafta Algoritmaların Analizi

Yinelemelerin çözümü

## Yinelemelerin çözümü

- Yinelemeler entegral, türev, v.s. denklemlerinin çözümlerine benzer. Yinelemeleri çözmek konusunda kullanacağımız 3 ana yöntem vardır. Bu yöntemler;
  - Yerine koyma metodu (substitution)
  - İterasyon (Yineleme) metodu (iteration metod)
    - Özyineleme ağacı (recursion tree)
  - Ana Metot (Master metod)
  - Ana yöntemlerin dışında tekrarlı bağıntılar Karakteristik denklemler kullanarak çözülebilir.

- Bazı tekrarlı bağıntıların çözümü yapılmadan çözümünün nasıl olabileceği hakkında tahmin yapılabilir. Çözüm tahmini yapılır ve yerine konulur. Yerine koyma (tahminler) metodu bir sınırdır ve tahmini ispatlamak için tümevarım yöntemi kullanılır.
- En genel yöntem:
  - 1. Çözümün şeklini tahmin edin.
  - o 2. Tümevarım ile doğrulayın.
  - o 3. Sabitleri çözün.
- o Örnek: T(n)=T(n/2)+c,  $n \ge 2$ , ve T(1)=1.
- $\circ$  T(2)=1+c, T(4)=1+2c, T(8)=1+3c, ...
- $\circ$  T(2<sup>k</sup>)=1+kc, burada n=2<sup>k</sup>, T(n)=1+clogn olur.

T(1)=1 verildi.

- Tümevarım ile İspat:
- Temel durum: n=1,
- Tümevarım hipotezi:  $n=2^k$ ,  $T(2^k)=1+kc$ , burada  $n=2^k$ , T(n)=T(n/2)+c için doğrudur.
- Tümevarım adımı:  $n = 2^{k+1}$ , T(n+1)=1+(k+1)\*c=(1+kc)+c
- ispat (proof):
- $T(n+1)=T(2^{k+1})=T(2^{k+1}/2)+c$
- $T(n+1) = T(2^{k*}2/2) + c$
- $=T(2^k) + c = 1 + kc + c$ , olur. Tümevarım adımı ispatlanmış olunur.
- O Burada, n=2<sup>k</sup> için, T(n)=1+clogn ve
- $T(n) \in O(\log n) \text{ dir.}$

- Ornek: T(n)=3T(n/2)+cn, n≥2, ve T(1)=1.
- $\circ$  T(2)=3+2c
- $\bullet$  T(4)=3(T(2))+4c=3(3+2c)+4c=9+10c=3<sup>2</sup>+[3<sup>1</sup>2<sup>1</sup>c]+3<sup>0</sup>2<sup>2</sup>c
- $\bullet$  T(8)=27+38c=3<sup>3</sup>+[3<sup>2</sup>2<sup>1</sup>c+3<sup>1</sup>2<sup>2</sup>c]+3<sup>0</sup>2<sup>3</sup>c
- T(16)=81+130c

$$f(n) = \sum_{0 \le i \le n} x^i = (x^{n+1}-1) / (x-1)$$

- **o** ...
- $T(2^k)=3^k+[3^{k-1}2^1c+3^{k-2}2^2c+...+3^12^{k-1}c]+3^02^kc$ ,  $2^kc$  parantezine alalım
- $T(2^k)=3^k+2^kc[(3/2)^{k-1}+(3/2)^{k-2}+...+(3/2)]$ , serisinin genel denklemi
- $T(2^k)=3^k+2^k c[((3/2)^k-1)/((3/2)-1)]$ , burada  $n=2^k$  ve k=logn
- T(n)= $3^{\log n}$ +cn[(( $3^{\log n}$ /n-1)/(1/2)], burada  $a^{\log_b x} = x^{\log_b a}$
- $\bullet$  T(n)=n<sup>log3</sup>+2cn(n<sup>log3-1</sup>-1) = n<sup>1,59</sup>+2cn(n<sup>0,59</sup>-1)
- $\bullet$  T(n)=n<sup>1,59</sup>(1+2c)-2cn
- $T(n) \in O(n^{1,59}) dir.$

- ispat: T(n)=3T(n/2)+cn,  $n\geq 2$ , ve T(1)=1. (1)
- $\bullet$  T(2<sup>k</sup>)=3<sup>k</sup>+2<sup>k</sup> c[((3/2)<sup>k</sup>-1)/((3/2)-1)], burada n=2<sup>k</sup> ve k=logn (2)
- Temel durum: n=1, T(1)=1 ve  $n=2 \rightarrow T(2)=3+2c$
- Tümevarım hipotezi:  $n=2^k \rightarrow T(2^k)=3^k+2^k c[((3/2)^k-1)/((3/2)-1)]$
- Tümevarım adımı:  $n=2^{k+1} \rightarrow T(2^{k+1})=3^{k+1}+2^{k+1} c[((3/2)^{k+1}-1)/((3/2)-1)]$  veya  $\rightarrow T(2^{k+1})=3^{k+1}+2^{k+2} c[(3/2)^{k+1}2-2]$  (3)
- Şimdi (1) nolu denklemi kullanarak (3) nolu denklemi ispatlayalım
- $T(2^k)=3^k.T(2^{k-1})+c2^k$  →  $n=2^k$  için
- $T(2^{k+1})=3.T(2^k)+c2^{k+1}=3.(3^k+c2^k[((3/2)^k-1)/((3/2)-1)])+c2^{k+1}$
- $T(2^{k+1})=3^{k+1}+2^k c[3.(3/2)^k-3)/(1/2)]+2^{k+1}c=3^{k+1}+2^{k+1} c[3.(3/2)^k-3]+2^{k+1}c$
- $T(2^{k+1})=3^{k+1}+2^{k+1}c([3.(3/2)^k-3]+1)=3^{k+1}+2^{k+1}c[(3/2)^{k+1}.2-2]$
- $T(2^{k+1})=3^{k+1}+2^{k+2}c[(3/2)^{k+1}2-2]$ , (3) nolu denklemi ispatlamış oluyoruz.
- $T(n) \in O(n^{1,59}) \text{ dir.}$

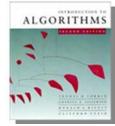
# Yerine koyma metodu (yöntemi) Üst sınırı tahmin ederek çözüm



• Örnek: T(n) = 4T(n/2) + n, ise

- $\circ$  [ T(1) =  $\Theta$ (1) olduğunu varsayın.]
- $\circ$  O(n<sup>3</sup>)'ü tahmin edin. (O ve Ω ayrı ayrı kanıtlayın.)
- k< n için T(k) ≤ ck³ olduğunu varsayın.</li>
- o T(n) ≤ cn³'ü tümevarımla kanıtlayın.

# Yerine koyma örneği



$$T(n) \le cn^{3}$$

$$T(n) = 4T(n/2) + n$$

$$\le 4c(n/2)^{3} + n$$

$$= (c/2)n^{3} + n$$

$$= cn^{3} - ((c/2)n^{3} - n) \leftarrow istenen - kalan$$

$$\le cn^{3} \leftarrow istenen$$

ne zaman ki 
$$(c/2)n^3 - n \ge 0$$
, örneğin, eğer  $c \ge 2$  ve  $n \ge 1$ .

# Yerine koyma örneği



- Başlangıç koşullarını da ele almalı, yani, tümevarımı taban şıklarına (base cases) dayandırmalıyız.
- 0
- **O Taban:**  $T(n) = \Theta(1)$  tüm  $n < n_0$  için, ki  $n_0$  uygun bir sabittir.
- 1 ≤ n <  $n_0$  için, elimizde " $\Theta(1)$ " ≤  $cn^3$ , olur; yeterince büyük bir c değeri seçersek.

Bu, sıkı bir sınır değildir!

# Yerine koyma örneği-Daha sıkı bir üst sınır?



$$T(n) = O(n^2)$$
 olduğunu kanıtlayacağız.

Varsayın ki  $T(k) \le ck^2$ , k < n için olsun :

$$T(n) = 4T(n/2) + n$$

$$\leq 4c(n/2)^{2} + n$$

$$= cn^{2} + n$$

$$= O(n^{2})$$

# Yerine koyma örneği-Daha sıkı bir üst sınır?



 $T(n) = O(n^2)$  olduğunu kanıtlayacağız.

Varsayın ki  $T(k) \le ck^2$ ; k < n: için←

$$T(n) = 4T(n/2) + n$$

$$\leq 4c(n/2)^{2} + n$$

$$= cn^{2} + n$$

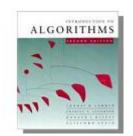
= (Yanlış! I.H.(tümevarım hipotezini) kanıtlamalıyız.

$$=cn^2-(-n)$$
 [istenen –kalan]

 $\leq cn^2$  seçeneksiz durum c > 0. Kaybettik!

-n >= 0 sağlanmaz (n değeri negatif olamaz)

# Yerine koyma örneği-Daha sıkı bir üst sınır?



Fikir: Varsayım hipotezini güçlendirin.

• Düşük-düzeyli bir terimi *çıkartın*.

*Varsayım hipotezi:*  $T(k) \le c_1 k^2 - c_2 k$ ;  $k \le n$  için.

$$T(n) = 4T(n/2) + n$$

$$= 4(c_1(n/2)^2 - c_2(n/2)) + n$$

$$= c_1n^2 - 2c_2n + n$$

$$= c_1n^2 - c_2n - (c_2n - n)$$

$$\leq c_1n^2 - c_2n \text{ eğer } c_2 \geq 1.$$

 $c_1$ 'i başlangıç koşullarını karşılayacak kadar büyük seçin.

- T(n)=3T(n/2)+cn,
- **o** Tahminimiz =T(k) $\in$  O(n<sup>1,59</sup>), n≥2, ve T(1)=1.
- İspatı yapabilmek için tahminimizi T(n) de yerine yazıp istediğimizi elde etmeye çalışmak. İstenen –kalan. Kalan, O'a eşit yada büyük olmalı
- $T(n) \le c_1 \cdot n^{1,59}$  olması
- $T(n)=3. (c_1 n^{1,59}/2^{1,59})+cn$
- $T(n)=c_1n^{1,59}-c_1n^{1,59}(2^{1,59}+3)/2^{1,59}+cn$
- istenen= $cn^{1,59}$ , Kalan=  $c_1n^{1,59}(2^{1,59}+3)/2^{1,59}+cn$
- istenen –kalan= $c_1 n^{1,59}$   $(c_1 n^{1,59} (2^{1,59} + 3)/2^{1,59} cn)$
- Kalan →  $c_1 n^{1,59} (2^{1,59} + 3)/2^{1,59} cn \ge 0$  sağlayacak c, ve n değeri var mı?
- **o** n=1 için,  $c_1(2^{1,59}+3)/2^{1,59}-c \ge 0$ ,  $c \ge 1$ , ve  $c_1 \ge 2$ , için sağlarız

#### Hanoi kulesi

```
void hanoi(int n, char source, char dest, char spare) {
                                                          Cost
 if (n > 0) {
                                                             c1
   hanoi(n-1, source, spare, dest);
                                                             c2
   cout << "Move top disk from pole " << source
                                                             c3
        << " to pole " << dest << endl;
   hanoi(n-1, spare, dest, source);
                                                             c4
  } }
when n=0
  T(0) = c1
when n>0
  T(n) = c1 + c2 + T(n-1) + c3 + c4 + T(n-1)
       = 2*T(n-1) + (c1+c2+c3+c4)
       function of hanoi-towers algorithm
```

- Hanoi kulesi
- $\circ$  T (n) = 2T (n 1) + 1 kabul edersek en iyi durumda T(0)=0 dır.
- $\bullet$  T (0) = 0, T (1) = 1, T (2) = 3, T (3) = 7, T (4) = 15, T (5) = 31, T (6) = 63, ...
- $\circ$  T(0)=0
- $\circ$  T(1)=2T(0)+1=1
- T(2)=2T(1)+1=2+1
- $\bullet$  T(3)=2T(2)+1=2.(2+1)+1=1+2+2<sup>2</sup>
- $\bullet$  T(4)=2T(3)+1=1+2+2<sup>2</sup>+2<sup>3</sup>
- $T(n)=2T(n-1)+1=1+2+...2^{n-1}$   $f(n) = \sum_{0 \le i \le n} 2^i = (2^{n+1}-1) / (2-1) = 2^{n+1}-1$
- $T(n)=2^{n}-1$
- o olur

# İterasyon metodu (Tümden gelim)

- Bu yöntemde verilen bağıntının çözümünü bulmak için büyük endeksli terimin yerine küçük endeksli terim yazılarak, genel terimin çözümü için bir yargıya varılıncaya kadar bu işleme devam edilir veya başlangıç şartlarına kadar devam edilir.
- İterasyon metodu, bir toplam içerisine yinelemeleri dönüştürür ve yinelemeleri çözmek için toplamları sınırlayıcı teknikleri kullanır.
  - Yineleme işlemini açık hale getir
  - Matematiksel işlemlerle göster
  - Toplamı hesapla.

• Örnek 
$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1\\ 2T(n/2) + \Theta(n) & \text{if } n > 1 \end{cases}$$

- o Örnek: Merge Sort, T(n)=2\*T(n/2)+n, n>1, ve T(1)=1 için yinelemeyi çözünüz
- o  $n=2^k \rightarrow k = log n hatırlayın,$

• 
$$T(n)=2*T(n/2)+n$$

substitute (yerine kullan)

• 
$$T(n)=2*(2*T(n/4)+n/2)+n$$

expand

• 
$$T(n)=2^2*T(n/4)+2*n$$

substitute

$$\bullet$$
 T(n)=2<sup>2</sup>\*(2\*T(n/8)+n/4)+2n =8\*T(n/8)+3\*n

expand

• 
$$T(n)=2^3*T(n/2^3)+3*n$$

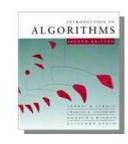
observe

• 
$$T(n)=2^{k*}T(n/n)+k*n$$

• 
$$T(n)=2^k * T(1)+k*n$$

• 
$$T(n)=\theta(n)+\theta(n\log n)$$

• 
$$T(n) \in \Theta(nlogn)$$



 Merge sort için daha kesin çalışma süresi bulma (bazı b değerleri için n=2<sup>b</sup> olduğunu varsayalım).

$$T(n) = \begin{cases} 2 & \text{if } n = 1\\ 2T(n/2) + 2n + 3 & \text{if } n > 1 \end{cases}$$

$$T(n) = 5n + 2nlgn - 3$$



$$T(n) = 2 T(n/2) + 2n + 3$$

$$= 2 (2 T(n/4) + n + 3) + 2n + 3$$

$$= 2^{2} T(n/4) + 4n + 2 \cdot 3 + 3$$

$$= 2^{2} T(n/2^{2}) + 2^{2}n + 2^{1} \cdot 3 + 2^{0} \cdot 3$$

$$= 2^{2} (2 T(n/8) + n/2 + 3) + 4n + 2 \cdot 3 + 3$$

$$= 2^{3} (T(n/2^{3}) + 2 \cdot 3n + (2^{2} + 2^{1} + 2^{0}) \cdot 3$$

$$= 2^{b} T(n/2^{b}) + 2 \cdot bn + 3 \sum_{j=0}^{b-1} 2^{j}$$

$$= n T(n/n) + 2n lgn + 3(2^{b} - 1)$$

$$= 2n + 2n lgn + 3n - 3$$

$$= 5n + 2n lgn - 3$$

- o Örnek: T(n)=T(n-1)+n, n>1, ve T(1)=1; için yinelemeyi çözünüz?
- 0
- T(n)=T(n-1) + n
- T(n)=T(n-2) + n-1+ n
- $\bullet$  T(n)=T(n-3) + n-2+ n-1+ n
- **o** ...
- T(n)=T(1) + 2 + .... n-2 + n-1 + n
- T(n)=1+2+.... n-2+n-1+n  $\longrightarrow f(n) = \sum_{1 \le i \le n} i = n(n+1)/2$
- T(n)=n\*(n+1)/2
- $T(n) \in \Theta(n^2)$

- Hanoi Kulesi
- $\circ$  T(n)=2\*T(n-1)+1, n>1, ve T(1)=1; için yinelemeyi çözünüz?
- Her bir hareket O(1) zaman gerektirir.

0

$$T(n)=2*T(n-1)+1=2^2*T(n-2)+2+1=....$$

$$\bullet$$
 T(n)=2<sup>n-1</sup> \* T(1)+2<sup>n-2</sup>+....+2+1

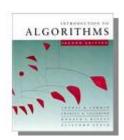
• 
$$T(n)=2^{n}-1$$

• 
$$T(n) \in \theta(2^n-1)$$

$$f(n) = \sum_{0 \le i \le n}^{\infty} 2^i = (2^{n+1}-1) / (2-1) = 2^{n+1}-1$$

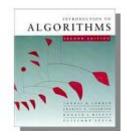
• n=64 için, her 1000 taşımanın 1 sn olduğu düşünülürse çözümü 584\*10<sup>6</sup> yıl alır.

# Özyineleme-ağacı metodu



- İterasyon yönteminin çözüm adımları ağaç şeklinde gösterilebilir ve elde edilen ağaca Özyineleme-ağacı denir
- Özyineleme-ağacı, bir algoritmadaki özyineleme uygulamasının maliyetini (zamanı) modeller.
- Özyineleme-ağacı metodu, diğer yöntemler gibi, güvenilir olmayabilir.
- Öte yandan özyineleme-ağacı metodu "öngörü" olgusunu geliştirir.
- Özyineleme-ağacı metodu "yerine koyma metodu" için gerekli tahminlerinde yararlıdır.

# Özyineleme-ağacı örneği



$$T(n) = T(n/4) + T(n/2) + n^{2}:$$

$$(n/4)^{2} \qquad (n/2)^{2} \qquad \frac{5}{16}n^{2}$$

$$(n/16)^{2} \qquad (n/8)^{2} \qquad (n/8)^{2} \qquad (n/4)^{2} \qquad \frac{25}{256}n^{2}$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$\Theta(1) \qquad \text{Total} = n^{2} \left(1 + \frac{5}{16} + \left(\frac{5}{16}\right)^{2} + \left(\frac{5}{16}\right)^{3} + \cdots\right)$$

$$= \Theta(n^{2}) \qquad \text{Geometrik seri}$$





```
T(n)
\Theta(1)
2T(n/2)
Suistimal
\Theta(n)
```

#### Birleştirme-Siralaması A[1 ... n]

- 1. Eğer n = 1'se, bitir.
- 2. Yinelemeli olarak  $A[1..\lceil n/2\rceil]$  ve  $A[\lceil n/2\rceil+1...n]$ 'yi sırala.
  - 3. 2 sıralı listeyi "Birleştir"

*Özensizlik:*  $T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor)$  olması gerekir, ama asimptotik açıdan bu önemli değildir.



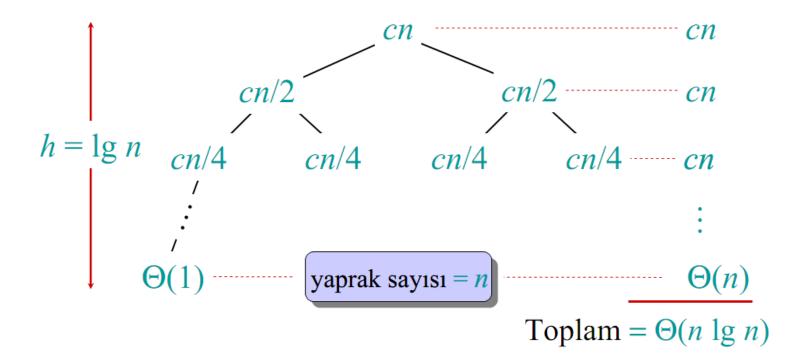


$$T(n) = \begin{cases} \Theta(1) \text{ eğer } n = 1 \text{ ise;} \\ 2T(n/2) + \Theta(n) \text{ eğer } n > 1 \text{ ise.} \end{cases}$$

 Genellikle n'nin küçük değerleri için taban durumu ( base case ) olan T(n) = Θ(1) 'i hesaplara katmayacağız; ama bunu sadece yinelemenin asimptotik çözümünü etkilemiyorsa yapacağız.

#### Yineleme ağacı

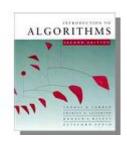
 $\circ$  T(n) = 2T(n/2) + cn'yi çözün; burada c > 0 bir sabittir.



## Sonuçlar- Insert Sort -Merge Sort

- Θ(n lg n)' nin, büyüme oranı Θ(n²)'den daha yavaştır (yani küçüktür).
- En kötü durumda, birleştirme sıralaması asimptotik olarak araya yerleştirme sıralamasından daha iyidir.
- Pratikte, birleştirme sıralaması(Merge Sort) araya yerleştirme sıralamasını (Insert Sort) n > 30 değerlerinde geçer.



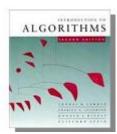


- Reküranslar değişken değiştirme ile daha basit hale dönüştürülebilir.
- $T(n) = 2T(\sqrt{n}) + \log n$ .  $m = \log n \Rightarrow 2^m = n \Rightarrow \sqrt{n} = 2^{m/2}$   $T(n) = 2T(\sqrt{n}) + \log n \Rightarrow T(2^m) = 2T(2^{m/2}) + m$  $S(m) = T(2^m)$

$$\begin{split} T(2^m) &= 2T(2^{m/2}) + m \ \Rightarrow S(m) = 2S(m/2) + m \\ &\Rightarrow S(m) = O(m\log m) \\ &\Rightarrow T(n) = T(2^m) = S(m) = O(m\log m) = O(\log n\log\log n) \end{split}$$

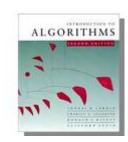
#### **Master Teorem**

#### **Ana Metod (The Master Method)**



- Ana method aşağıda belirtilen yapıdaki yinelemelere uygulanır:
- o T(n) = aT(n/b) + f(n), burada  $a \ge 1$ , b > 1, ve f asimptotik olarak pozitiftir.
- *T(n)* bir algoritmanın çalışma süresidir.
  - *n/b* boyutunda *a* tane alt problem recursive olarak çözülür ve her biri *T(n/b)* süresindedir.
  - *f(n)* problemin bölünmesi ve sonuçların birleştirilmesi için geçen süredir.
  - Örnek: Merge-sort için  $T(n)=2T(n/2)+\Theta(n)$  yazılabilir.

# Ana Metod (The Master Method) Üç yaygın uygulama



f(n)'i  $n^{\log_b a}$  ile karşılaştırın:

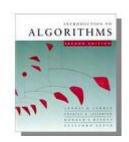
- 1.  $f(n) = O(n^{\log_b a \varepsilon})$   $\varepsilon > 0$  sabiti durumunda;
  - f(n) polinomsal olarak  $n^{\log_b a}$  göre daha yavaş büyür( $n^{\epsilon}$  faktörü oranında).

$$\mathbf{C\ddot{o}z\ddot{u}m}$$
:  $T(n) = \Theta(n^{\log_b a})$ .

- 2.  $f(n) = \Theta(n^{\log_b a} \lg^k n)$   $k \ge 0$  sabiti durumunda;
  - f(n) ve  $n^{\log_b a}$  benzer oranlarda büyürler.

$$\mathbf{C\ddot{o}z\ddot{u}m}$$
:  $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$ .

# Ana Metod (The Master Method) Üç yaygın uygulama

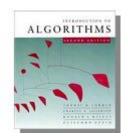


f(n)'i  $n^{\log_b a}$  ile karşılaştırın:

- 3.  $f(n) = \Omega(n^{\log_b a + \varepsilon})$   $\varepsilon > 0$  sabiti durumunda;
  - f(n) polinomsal olarak  $n^{\log_b a}$  'ye göre daha hızlı büyür ( $n^{\epsilon}$  faktörü oranında),

ve f(n), düzenlilik koşulunu  $af(n/b) \le cf(n)$  durumunda, c < 1 olmak kaydıyla karşılar. Çözüm:  $T(n) = \Theta(f(n))$ .

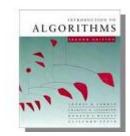
$$c=(1-\epsilon), \epsilon > 0$$



## Örnekler

Ör. 
$$T(n) = 4T(n/2) + n$$
  
 $a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2; f(n) = n.$   
Durum 1:  $f(n) = O(n^{2-\epsilon})$   $\epsilon = 1$  için.  
 $\therefore T(n) = \Theta(n^2).$ 

Ör. 
$$T(n) = 4T(n/2) + n^2$$
  
 $a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2; f(n) = n^2.$   
Durum 2:  $f(n) = \Theta(n^2 \lg^0 n)$ , yani,  $k = 0$ .  
 $\therefore T(n) = \Theta(n^2 \lg n)$ .

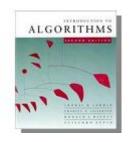


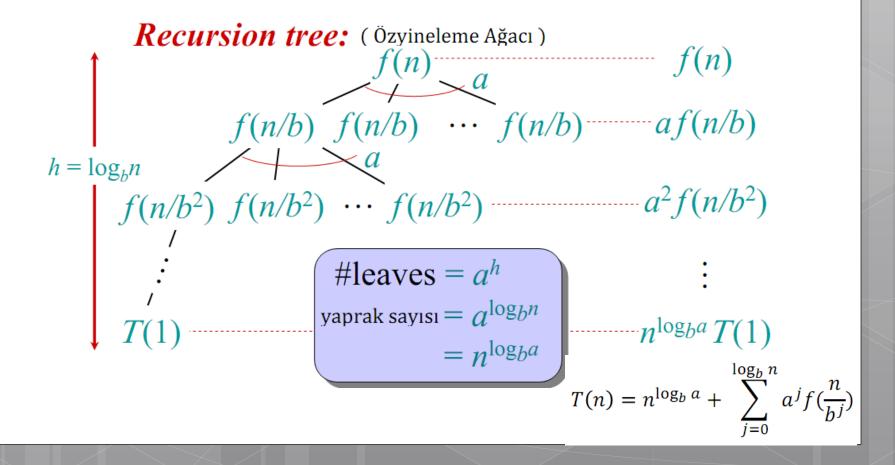
#### Örnekler

Ör. 
$$T(n) = 4T(n/2) + n^3$$
  
 $a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2; f(n) = n^3.$   
DURUM 3:  $f(n) = \Omega(n^{2+\epsilon})$   $\epsilon = 1$  için  
 $ve \ 4(n/2)^3 \le cn^3$  (düz. koş.)  $c = 1/2$  için  
 $\therefore T(n) = \Theta(n^3).$ 

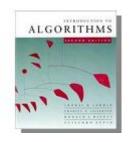
Ör. 
$$T(n) = 4T(n/2) + n^2/\lg n$$
  
 $a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2; f(n) = n^2/\lg n.$   
Ana metod geçerli değil. Özellikle,  
 $\varepsilon > 0$  olan sabitler için  $n^{\varepsilon} = \omega(\lg n)$  elde edilir.

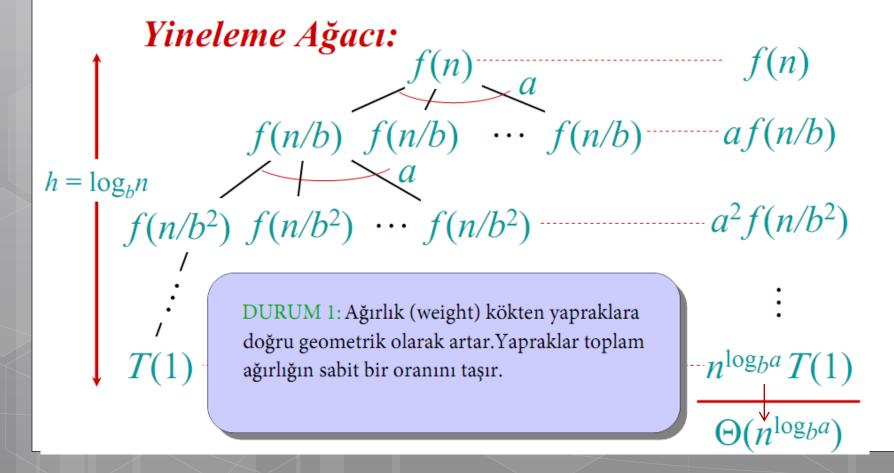
## Master teoremdeki düşünce



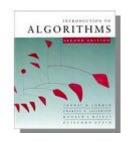


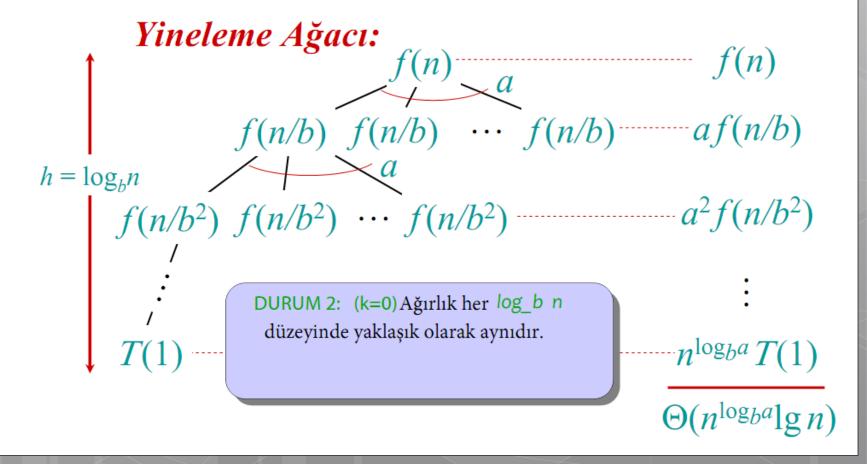
#### Master teoremdeki düşünce



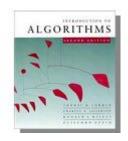


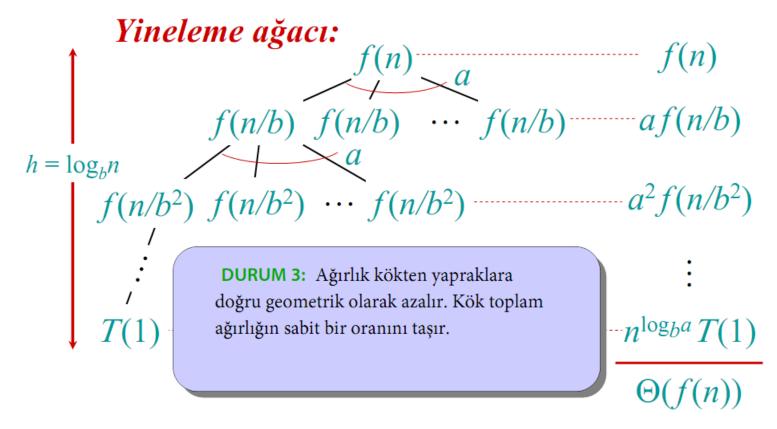
## Master teoremdeki düşünce



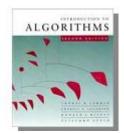


## Master teoremdeki düşünce





## Master teoremi ispat



- **Ourum 2:** Eğer  $f(n) = \theta(n^{\log_b a})$ , ise  $T(n) = \theta(n^{\log_b a} \log n)$
- ispat: Eğer  $f(n) = \theta(n^{\log_b a})$ , o  $zaman f(n) \le cn^{\log_b a}$  olur

$$T(n) = n^{\log_b a} + \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right)$$

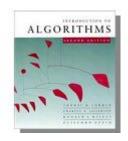
$$\bullet \le n^{\log_b a} + c \sum_{j=0}^{\log_b n-1} a^j \left(\frac{n}{b^j}\right)^{\log_b a}$$

$$\bullet = n^{\log_b a} + c n^{\log_b a} \sum_{j=0}^{\log_b n-1} a^j \left(\frac{1}{b^{\log_b a}}\right)^j a$$

$$\bullet = n^{\log_b a} + c n^{\log_b a} \sum_{j=0}^{\log_b a} 1 = n^{\log_b a} + c n^{\log_b a} \log_b n$$

- $\bullet \le c n^{\log_b a} \log n$
- Bu yüzden,  $f(n) = \theta(n^{\log_b a})$  ise  $T(n) = O(n^{\log_b a} \log n)$  dir.
- Durum 1 ve Durum 3 te benzerdir.( Önerilen ders kitabını inceleyiniz)

### Master teoremi



- o Örnek:  $T(n) = 9T\left(\frac{n}{3}\right) + n, n \ge 3 \ ve \ T(1) = 1$  ise çalışma zamanını bulunuz?
- Çözüm: a=9, b=3, f(n) = n ve  $n^{\log_b a} = n^{\log_3 9} = \theta(n^2)$
- **Ourum 1:**  $f(n) = O(n^{\log_3 9 \varepsilon})$ ,  $\varepsilon = 1$  için
- $T(n) = \theta(n^{\log_3 9}) = \theta(n^2)$

### Master teoremi



- o Örnek:  $T(n) = T\left(\frac{2n}{3}\right) + 1, n \ge 3$  ve T(2) = 1 ise çalışma zamanını bulunuz?
- o Çözüm: a=1, b=3/2 , f(n)=1 ve  $n^{log_ba}=n^{log_{3/2}1}=\theta(n^0)$
- Durum 2:  $f(n) = \Theta(n^{\log_b a} \lg^k n)$   $k \ge 0$  $T(n) = \theta(\log n)$

# ALGORITHMS

### Master teoremi

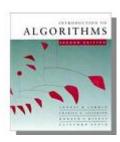
- o Örnek:  $T(n) = 3T(\frac{n}{4}) + nlogn, n \ge 4 \ ve \ T(1) = 1$  ise çalışma zamanını bulunuz?
- $\circ$  Çözüm: a=3, b=4, f(n) = nlogn ve

$$n^{\log_b a} = n^{\log_4 3} = O(n^{0.793})$$

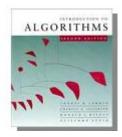
- **O Durum 3:**  $f(n) = \Omega(n^{\log_4 3 + \varepsilon})$ , burada  $\varepsilon \approx 0.2$  için düzenlilik koşulu,  $a*f(n/b) \le c*f(n)$ , büyük n değerleri ve c<1 olmak koşuluyla
- $\circ$  3(n/4)log(n/4)  $\leq$  (3/4) nlogn, c=3/4<1 için

$$T(n) = \theta(nlogn)$$

## Master teoremi ispat



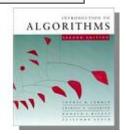
- **Ornek**:  $T(n) = 2T(\frac{n}{2}) + nlogn$  için master-metot durumları uygulanmaz?
- Burada a=2, b=2,  $n^{\log_b a} = n^{\log_2 2} = O(n)$  ve  $f(n) = n \log n$  dir.
- $\circ$  f(n), polinomsal olarak  $n^{log}{}_{b}{}^{a}$  göre hızlı büyüdüğünden **Durum 3** uygulanır.
- Büyüme oranı, asimptotik olarak çok büyük olmasına rağmen polinomsal olarak çok ta büyük değildir.
- Büyüme oranı  $f(n)/n^{\log_b a} = (n \log n)/n = \log n'$  dir. Bu oran herhangi bir pozitif  $\varepsilon$  sabiti için  $n^{\varepsilon}$ ' den asimptotik olarak azdır.
- Sonuç olarak çözüm Durum2 ve Durum3 arasına düşer.



- 1- $T(n) = T\left(\frac{n}{2}\right) + \sqrt{n} + \sqrt{60}46$  çözünüz.
- 2- $T(n) = T\left(\frac{n}{5}\right) + T\left(\frac{4n}{5}\right) + \theta(n)$  özyineleme ağacı kullanarak çözünüz.
- 3- Asimptotik notasyonlardan hangilerinin geçişme özelliği (transitivity) vardır.

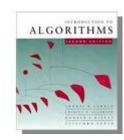
# ALGORITHMS

- 4- Aşağıdaki tekrarlı bağıntıların asimptotik davranışlarını inceleyiniz.(T(0)=1, T(1)=1)
  - a)  $T(n)=2T(n-1)-T(n-2)+\Theta(n)$
  - b)  $T(n)=3T(n-1)-2T(n-2)+\Theta(n2^n)$
  - c)  $T(n)=4T(n/2)-4T(n/4)+\Theta(n\lg n)$
  - d)  $T(n)=5T(n/2)-6T(n/4)+\Theta(n)$
- o 5- T(n)=T(√n/2√)+1 tekrarlı bağıntısı için T(n)=O(lgn) olduğunu gösteriniz.
- o 6- T(n)=T(n/2)+T(n/4)+T(n/8)+n öz yineleme ağacı ile çözünüz



- 7-  $T(n)=2T(\lfloor n/2 \rfloor)+n$  tekrarlı bağıntısı için  $T(n)=\Omega(n\lg n)$  ve  $T(n)=O(n\lg n)$  olduğunu gösteriniz. Son olarak  $T(n)=\Theta(n\lg n)$  asimptotik davranışı gösterip göstermediğini açıklayınız.
- 8- Aşağıdaki tekrarlı bağıntıyı çözünüz. (0<p<=q<1)</li>
  - $T(n)=T(pn)+T(qn)+\Theta(n)$
  - $\bullet$  T(1)= $\Theta$ (1)
- 9- i değişkeni 1 ile n-1 arasında değer alan düzenli dağıtık bir gelişigüzel değişken olsun.
- f(n)=(i+1)f(i) ve f(1)=1
   tekrarlı bağıntısı için f(n) nin mümkün olan değerleri nelerdir ve ortalama değeri nedir?
- 10- T(n) = 2 T( $\lfloor n/2 \rfloor$  + 17) + n tekrarlı bağıntısının çözümünün O(nlgn) olduğunu gösteriniz.

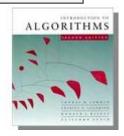
5.Hafta Karakteristik denklemler kullanarak yinelemeleri çözme



- 1- $T(n) = T\left(\frac{n}{2}\right) + \sqrt{n} + \sqrt{60}46$  çözünüz.
- 2- $T(n) = T\left(\frac{n}{5}\right) + T\left(\frac{4n}{5}\right) + \theta(n)$  özyineleme ağacı kullanarak çözünüz.
- 3-Aşağıdaki tekrarlı bağıntıların asimptotik davranışlarını inceleyiniz.(T(0)=1, T(1)=1)
  - a)  $T(n)=2T(n-1)-T(n-2)+\Theta(n)$
  - b)  $T(n)=3T(n-1)-2T(n-2)+\Theta(n2^n)$
  - c)  $T(n)=4T(n/2)-4T(n/4)+\Theta(n \lg n)$
  - d)  $T(n)=5T(n/2)-6T(n/4)+\Theta(n)$

# ALGORITHMS

- 4- T(n)=T(√n/2√)+1 tekrarlı bağıntısı için T(n)=O(lgn) olduğunu gösteriniz.
- o 5- T(n)=T(n/2)+T(n/4)+T(n/8)+n öz yineleme ağacı ile çözünüz
- 6-  $T(n)=2T(\lfloor n/2 \rfloor)+n$  tekrarlı bağıntısı için  $T(n)=\Omega(n\lg n)$  ve  $T(n)=O(n\lg n)$  olduğunu gösteriniz. Son olarak  $T(n)=\Theta(n\lg n)$  asimptotik davranışı gösterip göstermediğini açıklayınız.



- o 7- Aşağıdaki tekrarlı bağıntıyı çözünüz. (0<p<=q<1)</li>
  - $\bullet$  T(n)=T(pn)+T(qn)+ $\Theta$ (n)
  - $\bullet$  T(1)= $\Theta$ (1)
- 8- i değişkeni 1 ile n-1 arasında değer alan düzenli dağıtık bir gelişigüzel değişken olsun.
- f(n)=(i+1)f(i) ve f(1)=1 tekrarlı bağıntısı için f(n) nin mümkün olan değerleri nelerdir ve ortalama değeri nedir?
- 9- T(n) = 2 T( $\lfloor n/2 \rfloor$  + 17) + n tekrarlı bağıntısının çözümünün O(nlgn) olduğunu gösteriniz.

$$1 + x + x^{2} + \dots + x^{n} = \frac{1 - x^{n+1}}{1 - x} \quad ; x \neq 1 \quad icin$$

$$1 + x + x^{2} + \dots = \frac{1}{1 - x} \quad ; |x| < 1 \quad icin$$

$$S(N) = 1 + 2 + 3 + 4 + ... N = \sum_{i=1}^{N} i = \frac{N(N+1)}{2}$$

Karelerin Toplamı: 
$$\sum_{i=1}^{N} i^2 = \frac{N*(N+1)*(2n+1)}{6} \approx \frac{N^3}{3}$$

Geometrik Seriler: 
$$\sum_{i=0}^{N} A^{i} = \frac{A^{N+1} - 1}{A - 1}$$
 A > 1

$$\sum_{i=0}^{N} A^{i} = \frac{1 - A^{N+1}}{1 - A} = \Theta(1)$$
 |A| < 1

• İki sınır arasındaki sayıların toplamı:  $\sum_{i=a}^{b} f(i) = \sum_{i=0}^{b} f(i) - \sum_{i=0}^{a-1} f(i)$ 

$$\sum_{i=1}^{n} (4i^{2} - 6i) = 4\sum_{i=1}^{n} i^{2} - 6\sum_{i=1}^{n} i$$

- Combinatorics
  - **1.** Number of permutations of an *n*-element set: P(n) = n!
  - **2.** Number of *k*-combinations of an *n*-element set:  $C(n, k) = \frac{n!}{k!(n-k)!}$
  - 3. Number of subsets of an *n*-element set:  $2^n$

Properties of Logarithms

1. 
$$\log_a 1 = 0$$

**2.** 
$$\log_a a = 1$$

$$3. \quad \log_a x^y = y \log_a x$$

$$4. \quad \log_a xy = \log_a x + \log_a y$$

$$5. \quad \log_a \frac{x}{y} = \log_a x - \log_a y$$

$$6. \quad a^{\log_b x} = x^{\log_b a}$$

7. 
$$\log_a x = \frac{\log_b x}{\log_b a} = \log_a b \log_b x$$

## Bazı Önemli Matematiksel İfadeler

#### Important Summation Formulas

**1.** 
$$\sum_{i=l}^{u} 1 = \underbrace{1 + 1 + \dots + 1}_{u-l+1 \text{ times}} = u - l + 1 \ (l, u \text{ are integer limits}, l \le u); \sum_{i=1}^{n} 1 = n$$

2. 
$$\sum_{i=1}^{n} i = 1 + 2 + \dots + n = \frac{n(n+1)}{2} \approx \frac{1}{2}n^2$$

3. 
$$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6} \approx \frac{1}{3}n^3$$

**4.** 
$$\sum_{i=1}^{n} i^{k} = 1^{k} + 2^{k} + \dots + n^{k} \approx \frac{1}{k+1} n^{k+1}$$

5. 
$$\sum_{i=0}^{n} a^{i} = 1 + a + \dots + a^{n} = \frac{a^{n+1} - 1}{a - 1} \ (a \neq 1); \quad \sum_{i=0}^{n} 2^{i} = 2^{n+1} - 1$$

**6.** 
$$\sum_{i=1}^{n} i 2^{i} = 1 \cdot 2 + 2 \cdot 2^{2} + \dots + n 2^{n} = (n-1)2^{n+1} + 2$$
 
$$\sum_{i=0}^{n-1} i x^{i} = x + 2x^{2} + 3x^{3} \dots + n x^{n} = \frac{(n-1)x^{(n+1)} - n x^{n} + x}{(x-1)^{2}}.$$

7. 
$$\sum_{i=1}^{n} \frac{1}{i} = 1 + \frac{1}{2} + \dots + \frac{1}{n} \approx \ln n + \gamma$$
, where  $\gamma \approx 0.5772 \dots$  (Euler's constant)

$$8. \quad \sum_{i=1}^{n} \lg i \approx n \lg n$$

## Bazı Önemli Matematiksel İfadeler

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

2. 
$$\sum_{k=1}^{n} 2k = 2+4+6+...+2n = n(n+1)$$

3. 
$$\sum_{k=1}^{n} (2k-1) = 1+3+5+...+(2n-1)=n^{2}$$

4. 
$$\sum_{k=1}^{n} k^2 = 1^2 + 2^2 + 3^2 + ... + n^2 = \frac{n(n+1)(2n+1)}{6}$$

5. 
$$\sum_{k=1}^{n} k^3 = 1^3 + 2^3 + 3^3 + ... + n^3 = \left[\frac{n(n+1)}{2}\right]^2$$

6. 
$$\sum_{k=1}^{n} r^{k-1} = 1 + r + r^2 + r^3 + ... + r^{n-1} = \frac{1 - r^n}{1 - r}, \ (r \neq 1)$$

7. 
$$\sum_{k=1}^{n} \frac{1}{k \cdot (k+1)} = \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + ... + \frac{1}{n \cdot (n+1)} = \frac{n}{n+1}$$

8. 
$$\sum_{k=1}^{n} k \cdot k! = (n+1)! - 1$$

Sum Manipulation Rules

1. 
$$\sum_{i=1}^{u} ca_i = c \sum_{i=1}^{u} a_i$$

**2.** 
$$\sum_{i=l}^{u} (a_i \pm b_i) = \sum_{i=l}^{u} a_i \pm \sum_{i=l}^{u} b_i$$

3. 
$$\sum_{i=l}^{u} a_i = \sum_{i=l}^{m} a_i + \sum_{i=m+1}^{u} a_i$$
, where  $l \le m < u$ 

**4.** 
$$\sum_{i=l}^{u} (a_i - a_{i-1}) = a_u - a_{l-1}$$

#### Approximation of a Sum by a Definite Integral

$$\int_{l-1}^{u} f(x)dx \le \sum_{i=l}^{u} f(i) \le \int_{l}^{u+1} f(x)dx \quad \text{for a nondecreasing } f(x)$$

$$\int_{l}^{u+1} f(x)dx \le \sum_{i=l}^{u} f(i) \le \int_{l-1}^{u} f(x)dx \quad \text{for a nonincreasing } f(x)$$

#### Floor and Ceiling Formulas

The *floor* of a real number x, denoted  $\lfloor x \rfloor$ , is defined as the greatest integer not larger than x (e.g.,  $\lfloor 3.8 \rfloor = 3$ ,  $\lfloor -3.8 \rfloor = -4$ ,  $\lfloor 3 \rfloor = 3$ ). The *ceiling* of a real number x, denoted  $\lceil x \rceil$ , is defined as the smallest integer not smaller than x (e.g.,  $\lceil 3.8 \rceil = 4$ ,  $\lceil -3.8 \rceil = -3$ ,  $\lceil 3 \rceil = 3$ ).

- 1.  $x 1 < \lfloor x \rfloor \le x \le \lceil x \rceil < x + 1$
- **2.**  $\lfloor x + n \rfloor = \lfloor x \rfloor + n$  and  $\lceil x + n \rceil = \lceil x \rceil + n$  for real x and integer n
- 3.  $\lfloor n/2 \rfloor + \lceil n/2 \rceil = n$
- **4.**  $\lceil \lg(n+1) \rceil = \lfloor \lg n \rfloor + 1$

#### Miscellaneous

- 1.  $n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$  as  $n \to \infty$  (Stirling's formula)
- **2.** Modular arithmetic (n, m are integers, p is a positive integer)

$$(n+m) \mod p = (n \mod p + m \mod p) \mod p$$
  
 $(nm) \mod p = ((n \mod p)(m \mod p)) \mod p$