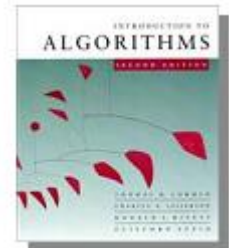




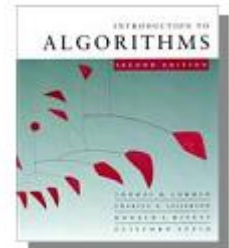
3.Hafta

Master Teorem ve Böl-Fethet Metodu



Ana Metod (The Master Method)

- Ana method aşağıda belirtilen yapıdaki yinelemelere uygulanır:
- $T(n) = aT(n/b) + f(n)$, burada $a \geq 1$, $b > 1$, 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

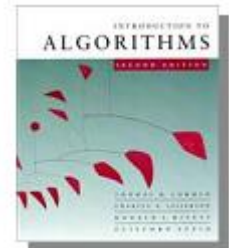
$f(n)$ 'i $n^{\log_b a}$ ile karşılaştırın:

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^ε faktörü oranında).

Çözüm: $T(n) = \Theta(n^{\log_b a})$.

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

Çözüm: $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$.



Ana Metod (The Master Method)

Üç yaygın uygulama

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

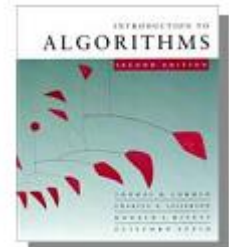
3. $f(n) = \Omega(n^{\log_b a + \varepsilon})$ $\varepsilon > 0$ sabiti durumunda;

- $f(n)$ polinomsal olarak $n^{\log_b a}$ 'ye göre daha hızlı büyür (n^ε faktörü oranında),

ve $f(n)$, **düzenlilik koşulunu** $af(n/b) \leq cf(n)$ durumunda, $c < 1$ olmak kaydıyla karşılar.

Çözüm: $T(n) = \Theta(f(n))$.

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



Örnekler

Ör. $T(n) = 4T(n/2) + n$

$$a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2; f(n) = n.$$

Durum 1: $f(n) = O(n^{2-\varepsilon})$ $\varepsilon = 1$ için.

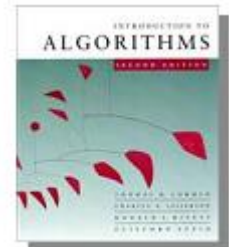
$$\therefore T(n) = \Theta(n^2).$$

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

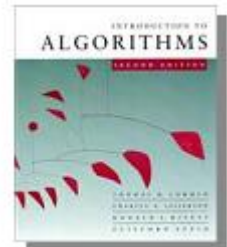
$$a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2; f(n) = n^3.$$

DURUM 3: $f(n) = \Omega(n^{2+\varepsilon})$ $\varepsilon = 1$ için
 ve $4(n/2)^3 \leq cn^3$ (düz. koş.) $c = 1/2$ için
 $\therefore T(n) = \Theta(n^3)$.

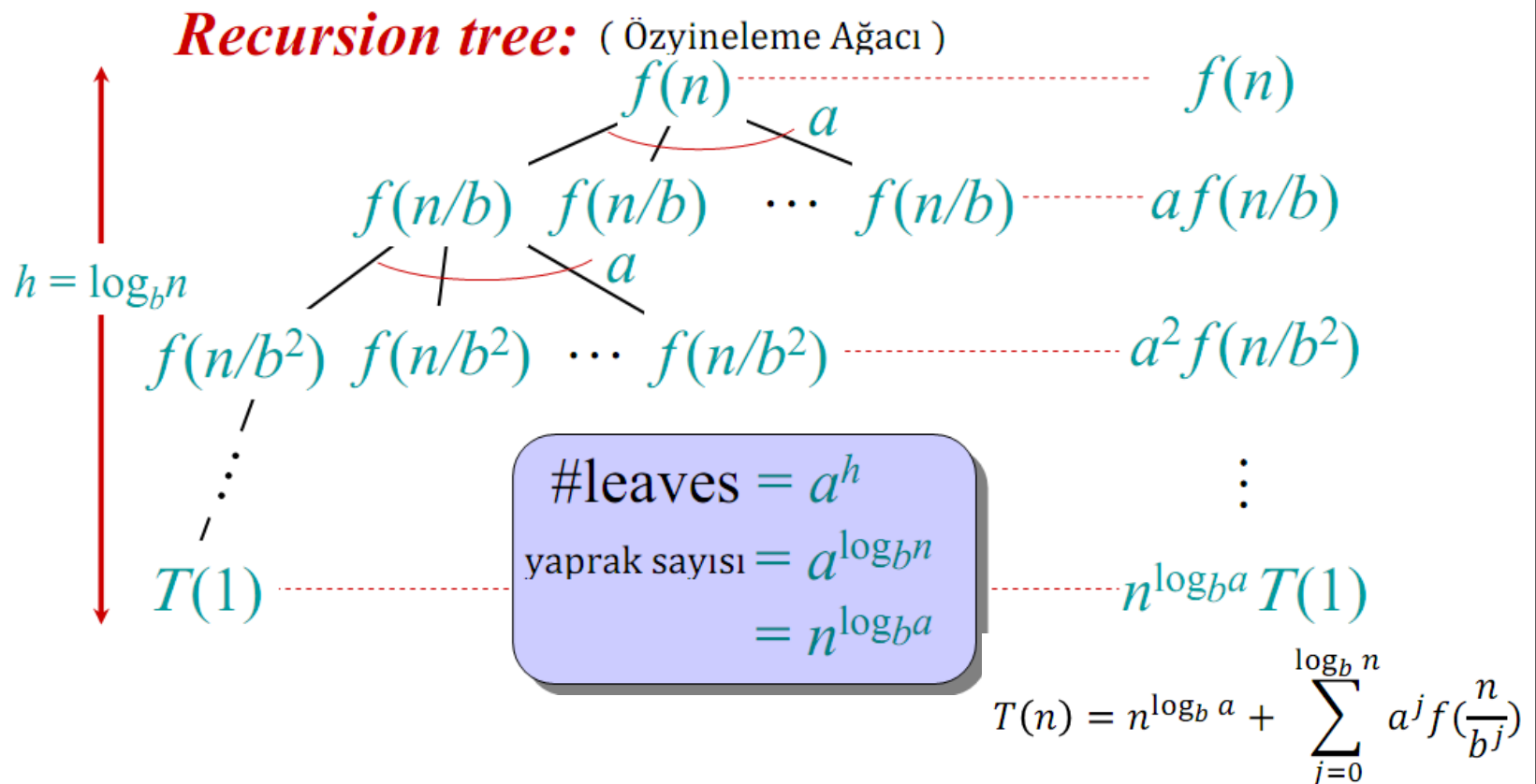
Ör. $T(n) = 4T(n/2) + n^2/\lg n$

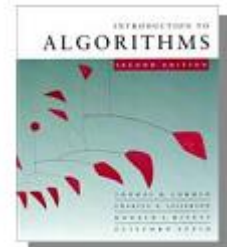
$$a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2; f(n) = n^2/\lg n.$$

Ana metod geçerli değil. Özellikle,
 $\varepsilon > 0$ olan sabitler için $n^\varepsilon = \omega(\lg n)$ elde edilir.



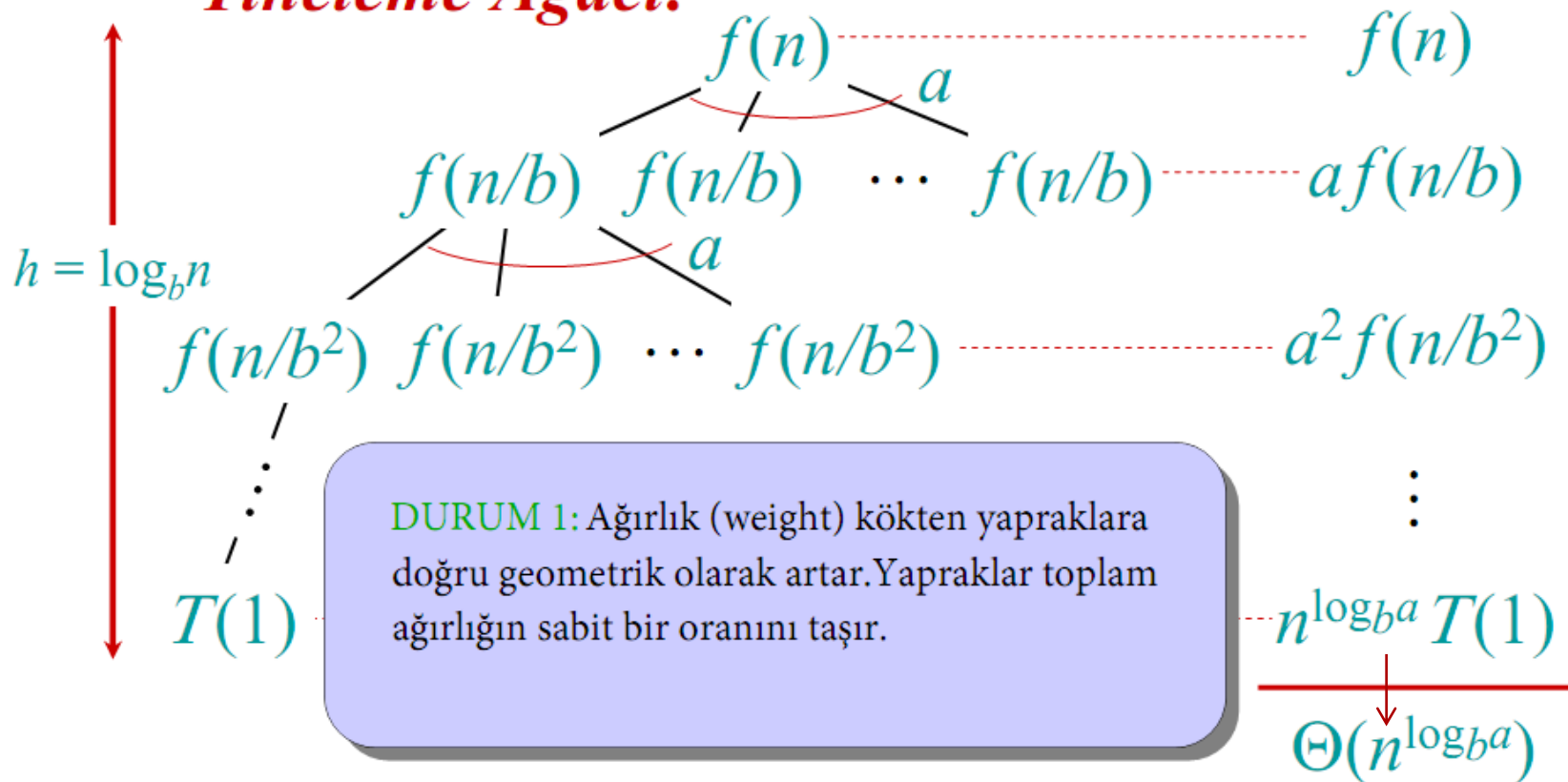
Master teoremdaki düşünce

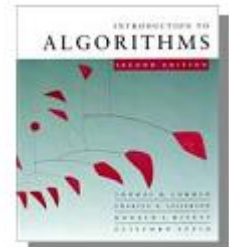




Master teoremdaki düşünce

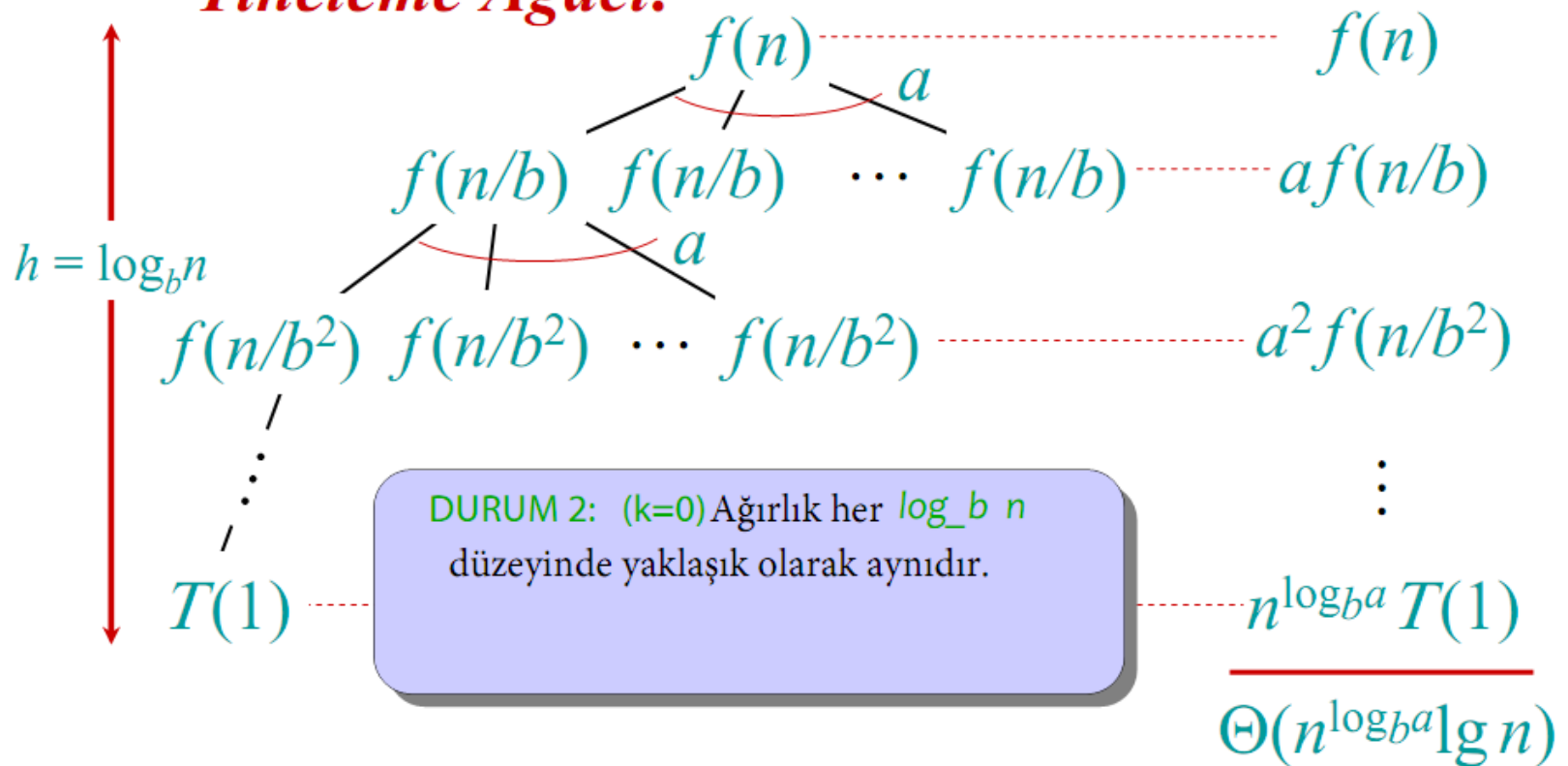
Yineleme Ağacı:

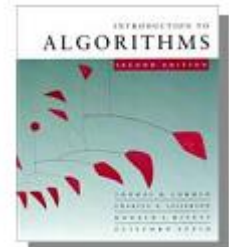




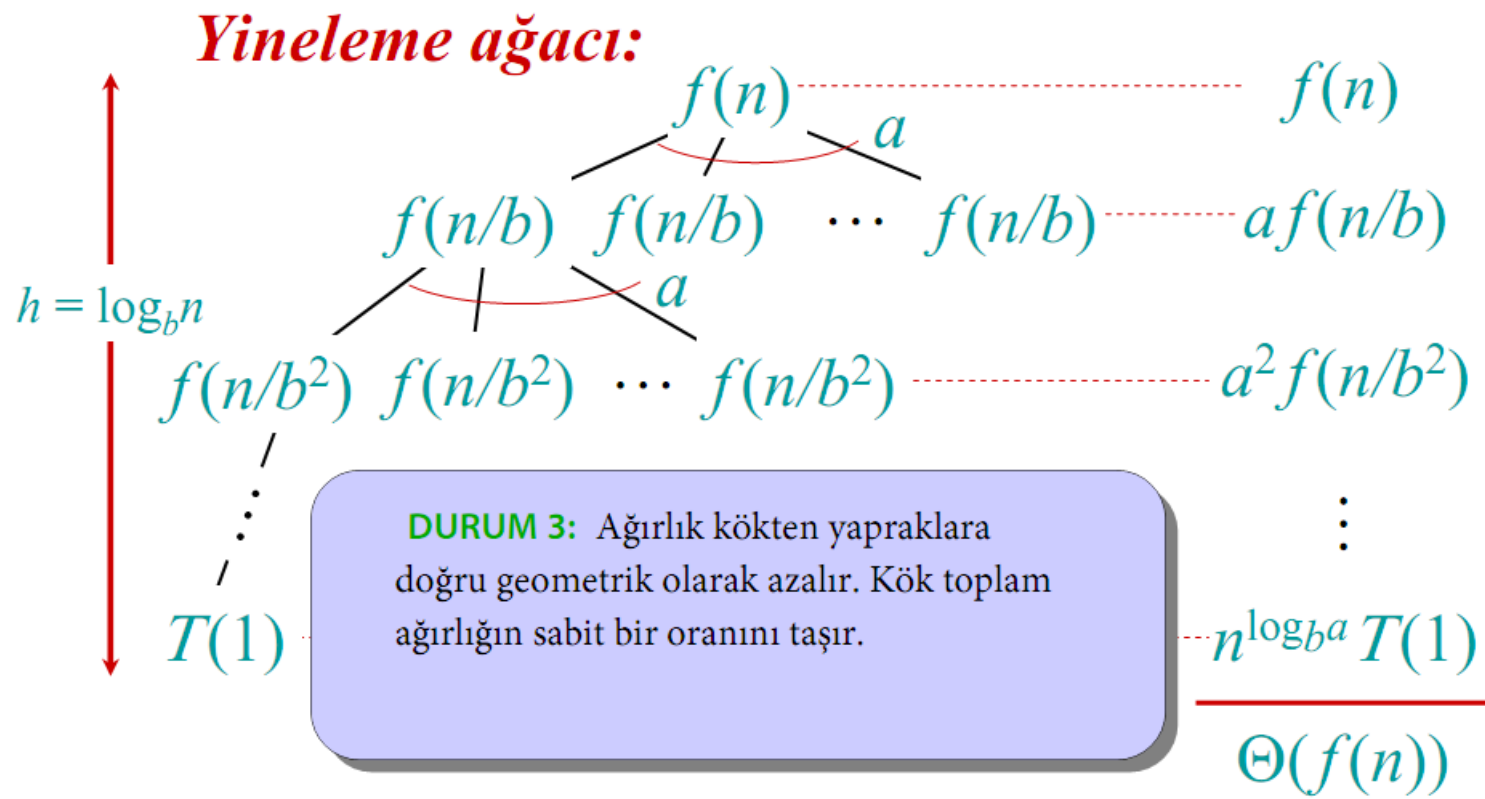
Master teoremdaki düşünce

Yineleme Ağacı:

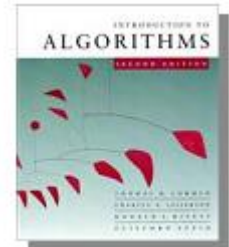




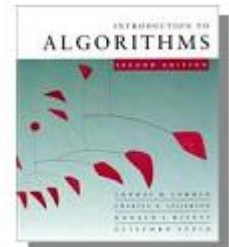
Master teoremdaki düşünce



Master teoremi ispat

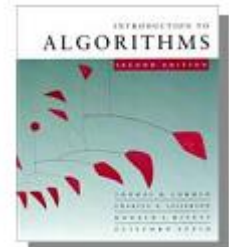


- **Durum 2:** Eğer $f(n) = \theta(n^{\log_b a})$, ise $T(n) = \theta(n^{\log_b a} \log n)$
- **İspat:** Eğer $f(n) = \theta(n^{\log_b a})$, o zaman $f(n) \leq cn^{\log_b a}$ olur
- $$T(n) = n^{\log_b a} + \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right)$$
- $$\leq n^{\log_b a} + c \sum_{j=0}^{\log_b n - 1} a^j \left(\frac{n}{b^j}\right)^{\log_b a}$$
- $$= n^{\log_b a} + cn^{\log_b a} \sum_{j=0}^{\log_b n - 1} a^j \left(\frac{1}{b^{\log_b a}}\right)^j$$
- $$= n^{\log_b a} + cn^{\log_b a} \sum_{j=0}^{\log_b n - 1} 1 = n^{\log_b a} + cn^{\log_b a} \log_b n$$
- $$\leq cn^{\log_b a} \log n$$
- 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)



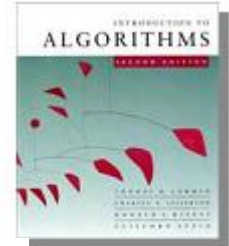
Master teoremi

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



Master teoremi

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



Master teoremi

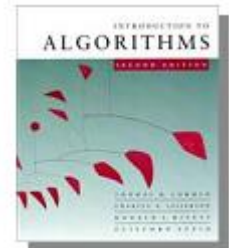
- **Örnek:** $T(n) = 3T\left(\frac{n}{4}\right) + n \log n, n \geq 4$ ve $T(1) = 1$ ise çalışma zamanını bulunuz?

- **Çözüm:** $a=3, b=4, f(n) = n \log n$ ve

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

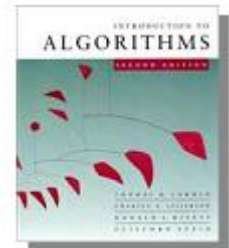
- **Durum 3:** $f(n) = \Omega(n^{\log_4 3 + \varepsilon})$, burada $\varepsilon \approx 0.2$ için düzenlilik koşulu, $a \cdot f(n/b) \leq c \cdot f(n)$, büyük n değerleri ve $c < 1$ olmak koşuluyla
- $3(n/4) \log(n/4) \leq (3/4) n \log n, c=3/4 < 1$ için

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



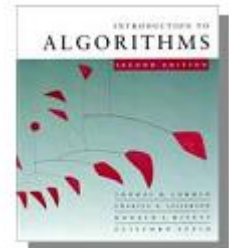
Master teoremi ispat

- ❶ Örnek: $T(n) = 2T\left(\frac{n}{2}\right) + n \log n$ 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.
- $f(n)$, 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 ε sabiti için n^{ε} den asimptotik olarak azdır.
- Sonuç olarak çözüm Durum2 ve Durum3 arasına düşer.



Karakteristik denklemler kullanarak yinelemeleri çözme

- Doğrusal Yineleme (Rekürans) Bağıntısı
- Bir yinelemeli bağıntıda t_n , dizinin önceki terimlerinin katlarının toplamına eşitse doğrusal (lineer) dir. ($t_n \rightarrow T(n)$)
 - $t_n = t_{n-1} + t_{n-2}$ doğrusal
 - $t_n = t_{n-1} + t_{n-2}^2$ doğrusal değildir, t_{n-2}^2 önceki terimin katı değildir.
- Homojen Yineleme (Rekürans) Bağıntısı:
- t_n sadece önceki terimlerin katlarına bağlı ise homojen (türdeş) olarak adlandırılır.
 - $t_n = t_{n-1} + t_{n-2}$ homojen
 - $t_n = 2t_{n-1} + 1$ homojen değildir. "+1" terimi t_j katı değildir.

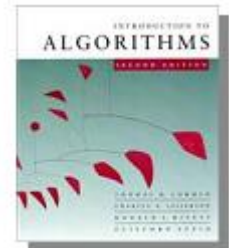


Karakteristik denklemler kullanarak özyinelemeleri çözme

- Yinelemeli bağıntıdaki terimlerin katsayıları sabit ise; sabit katsayılı homojen doğrusal yineleme formu aşağıdaki gibidir.

$$c_0 t_n + c_1 t_{n-1} + \dots + c_k t_{n-k} = 0$$

- Burada,
 - t_i : özyinelemeli bağıntının değerlerini,
 - c_i : sabit katsayılı terimlerini ifade eder.
 - c_i , reel sayılardır ve $c_i \neq 0$.
 - k : ise özyinelemeli bağıntının derecesidir.



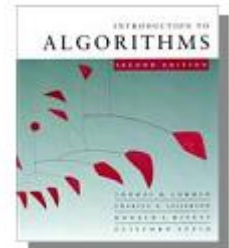
Karakteristik denklemler kullanarak özyinelemeleri çözme

- Doğrusal özyinelemelerde t_{i+j} , t_i^2 şeklinde terimler bulunmaz.
- Öz yineleme homojendir, çünkü t_i nin doğrusal kombinasyonundan dolayı 0 (sıfır)' a eşittir.
- Bu öz yinelemeler k başlangıç koşullarını içerir.

$$t_n = c_0 \quad t_1 = c_1 \quad \dots \quad t_k = c_k$$

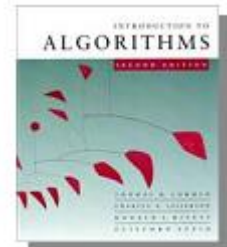
- Fibonacci dizisi için özyineleme

- $f_n = f_{n-1} + f_{n-2} \rightarrow f_n - f_{n-1} - f_{n-2} = 0$,
burada $k=2$, $c_0=1$ ve $c_1 = c_2 = -1$ dir.



Karakteristik denklemler kullanarak özyinelemeleri çözme

- Sabit katsayılı homojen doğrusal yineleme bağıntılarını çözmek için basit bir yöntem vardır. Bu yöntem;
 - k bir sabit olmak üzere, $t_k = x^k$;
 - $t_n = c_1 t_{n-1} + c_2 t_{n-2} + \dots + c_k t_{n-k}$ 'nin bir çözümü kabul edilir ve bağıntıda yerine konulursa
 - $x^n = c_1 x^{n-1} + c_2 x^{n-2} + \dots + c_k x^{n-k}$ elde edilir. Burada, x bilinmeyen bir sabit ve $x \neq 0$ dır.
- Bu ifadenin her iki yanını x^{n-k} ile bölersek:
 - $x^k - c_1 x^{k-1} - c_2 x^{k-2} - \dots - c_k = 0$ bulunur ve derecesi k olan ve genelde k adet kökü olan bu polinoma yineleme bağıntısının **karakteristik denklemi** (characteristic equation) adı verilir. Bu denklemin kökü birden fazla veya karmaşık olabilir.



Karakteristik denklemler kullanarak özyinelemeleri çözme

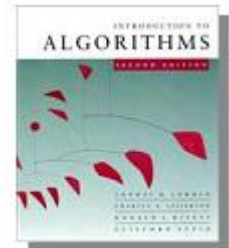
İşlem Adımları

Adım 1

- $x^2 - c_1x - c_2 = 0$ karakteristik denklemi için
- İkinci dereceden bir denklem olduğundan karakteristik kökleri r_1 ve r_2 olup
- $$x_{1,2} = \frac{c_1 \pm \sqrt{(c_1)^2 + 4c_2}}{2}$$

Adım 2

- **Durum 1:** Köklerin hiç biri aynı değilse
- $t_n = c_1x_1^n + c_2x_2^n$
- **Durum 2:** Köklerde aynı olan değer var ise ($x_1 = x_2$)
- $t_n = c_1x_0^n + c_2x_1^n + c_3nx_1^n$



Karakteristik denklemler kullanarak öz yinelemeleri çözme

Adım 3

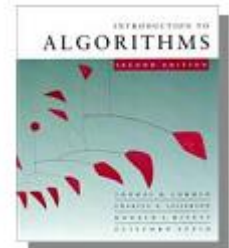
- Bir önceki adımda elde edilen denklemlere ilk koşulları uygulayınız.
- Durum 1: Kökler eşit değil
- $t_0 = c_1 x_1^0 + c_2 x_2^0 = c_1 + c_2$
- $t_1 = c_1 x_1^1 + c_2 x_2^1 = c_1 x_1 + c_2 x_2$
- Durum 2: Kökler eşit ($x_2 = x_1 = x_0$)
- $t_0 = c_1 x_0^0 + c_2 \cdot 0 \cdot x_0^0 = c_1$
- $t_1 = c_1 x_0^1 + c_2 \cdot 1 \cdot x_0^1 = (c_1 + c_2) x_0$

Adım 4

- c_1, c_2 'yi bulunuz

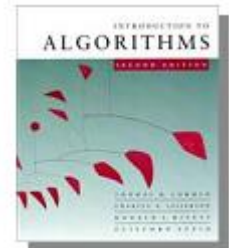
Adım 5

t_n için genel çözümü yaz



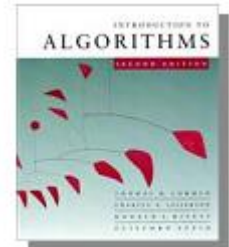
Karakteristik denklemler kullanarak özyinelemeleri çözme

- **Örnek:** İlk koşullar $t_0 = 2$ ve $t_1 = 7$ olarak verildiğine göre
- $t_n = t_{n-1} + 2t_{n-2}$ yinelemeli bağıntıyı çözünüz
- Karakteristik denklem: $x^2 - x - 2 = 0$
- Kökler $x_1 = 2$ ve $x_2 = -1$, kökler eşit değil. Durum 1'i kullanılacak.
- $t_n = c_1 2^n + c_2 (-1)^n$
- $t_0 = 2 = c_1 + c_2$, $t_1 = 7 = c_1 \cdot 2 + c_2 \cdot (-1)$
- $c_1 = 3$ ve $c_2 = -1$ olarak bulunur.
- Bu değerleri yerine yazarsak
- $t_n = 3 \cdot 2^n + (-1) \cdot (-1)^n = 3 \cdot 2^n - (-1)^n$ olarak bulunur.
- $t_n \in \theta(2^n)$



Karakteristik denklemler kullanarak özyinelemeleri çözme

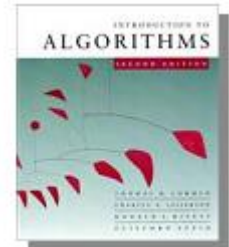
- ➊ **Örnek:** İlk koşullar $t_0 = 0$ ve $t_1 = 1$ ve $n \geq 2$ olarak verildiğine göre
- $t_n - 3t_{n-1} - 4t_{n-2} = 0$ için yinelemeli bağıntıyı çözünüz
- Karakteristik denklem: $x^2 - 3x - 4 = 0$
- Kökler $x_1 = -1$ ve $x_2 = 4$, kökler eşit değil. Durum 1'i kullanılacak.
- $t_n = c_1(-1)^n + c_2 4^n$
- $t_0 = 0 = c_1 + c_2, t_1 = 1 = c_1 \cdot (-1) + 4c_2$
- $c_1 = -1/5$ ve $c_2 = 1/5$ olarak bulunur.
- Bu değerleri yerine yazarsak
- $t_n = 1/5[4^n - (-1)^n]$ olarak bulunur.
- $t_n \in \theta(4^n)$



Karakteristik denklemler kullanarak özyinelemeleri çözme

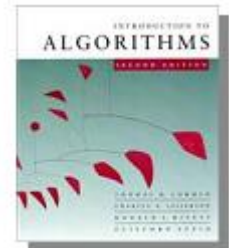
🌀 **Örnek:** Fibonacci özyinelemeli bağıntı

- $t_0 = 0$ ve $t_1 = 1$ ve $n \geq 2$ olarak verildiğine göre
- $t_n - t_{n-1} - t_{n-2} = 0$ için yinelemeli bağıntıyı çözünüz
- Karakteristik denklem: $x^2 - x - 1 = 0$
- Kökler $x_1 = \frac{1+\sqrt{5}}{2}$ ve $x_2 = \frac{1-\sqrt{5}}{2}$, kökler eşit değil. ($\frac{1+\sqrt{5}}{2}$, altın oran)
- Durum 1'i kullanılacak. $t_n = c_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + c_2 \left(\frac{1-\sqrt{5}}{2}\right)^n$
- $t_n(0) = 0 = c_1 + c_2$, $t_1 = 1 = c_1 \cdot \frac{1+\sqrt{5}}{2} + c_2 \cdot \frac{1-\sqrt{5}}{2}$
- $c_1 = 1/\sqrt{5}$ ve $c_2 = -1/\sqrt{5}$ olarak bulunur.
- Bu değerleri yerine yazarsak
- $t_n = 1/\sqrt{5} \left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n \right]$ olarak bulunur ve sonuç olarak
- $t_n \in \theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$



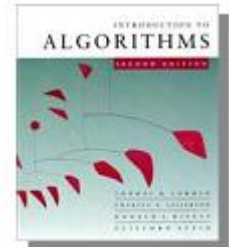
Karakteristik denklemler kullanarak özyinelemeleri çözme

- **Örnek:** $t_0 = 0, t_1 = 1$ ve $t_2 = 2$ ve $n \geq 3$ olarak verildiğine göre
- $t_n = 5t_{n-1} - 8t_{n-2} + 4t_{n-3}$ için yinelemeli bağıntıyı çözünüz
- Karakteristik denklem: $x^3 - 5x^2 + 8x - 4 = 0$, veya $(x-1)(x-2)^2$
- Kökler $x_1 = 1$, ve $x_2 = x_3 = 2$, (iki kök eşit). Eşit kökler bulunduğu için Durum 2 kullanılacak
- $t_n = c_1(1)^n + c_2(2)^n + c_3n(2)^n$
- Başlangıç şartlarına göre;
- $c_1 + c_2 = 0; (n = 0)$,
- $c_1 + 2c_2 + 2c_3 = 1; (n = 1)$,
- $c_1 + 4c_2 + 8c_3 = 2; (n = 2)$,
- $c_1 = -2, c_2 = 2$ ve $c_3 = -1/2$ olarak bulunur.
- Bu değerleri yerine yazarsak
- $t_n = 2^{n+1} - n2^{n-1} - 2$ olarak bulunur.



Karakteristik denklemler kullanarak yinelemeleri çözme

- **Homojen Olmayan Yineleme Bağlılıkları**
- t_n sadece önceki terimlerin katlarına bağlı değil ise homojen olmayan bağıntı olarak adlandırılır.
 - $t_n = t_{n-1} + t_{n-2}$ homojen
 - $t_n = 2t_{n-1} + 1$ homojen değildir. "+1" terimi t_j katı değildir.
- Yinelemeli bağıntıların genel formu $c_0 t_n + c_1 t_{n-1} + \dots + c_k t_{n-k} = f(n)$ şeklinde ifade edilir. $f(n) = 0$ eşit ise homojen, sıfırdan farklı ise homojen olmayan yinelemeli bağıntıdır.
- **$f(n) = b^n p(n)$**
- şeklinde ifade edilirse b sıfırdan farklı bir sabiti $p(n)$ ise d . dereceden n nin bir polinomudur.



Karakteristik denklemler kullanarak yinelemeleri çözme

- **Örnek:** Aşağıda verilen reküransı çözünüz
- $t_n - 2t_{n-1} = 3^n$ burada $b=3$, $p(n)^d=1$ ve polinom derecesi $d=0$ 'dır.

- İlk olarak her iki tarafı 3 ile çarpalım:

$$3t_n - 6t_{n-1} = 3^{n+1}$$

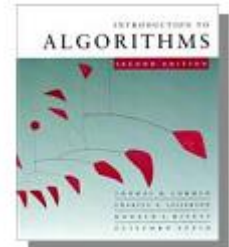
- Eğer n , $n+1$ ile yer değiştirirsek:

$$t_{n+1} - 2t_n = 3^{n+1} \quad \text{denklemini elde ederiz.}$$

- Her iki denklemi bir birinden çıkarırsak yeni denklem

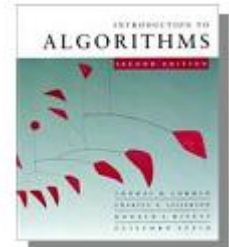
- $t_{n+1} - 5t_n + 6t_{n-1} = 0$ olur.

Karakteristik denklemler kullanarak yinelemeleri çözme



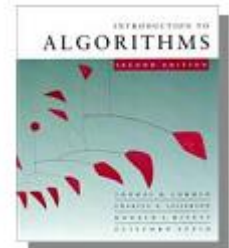
- Homojen durumda olduğu gibi çözüm yaparsak karakteristik denklem
- $x^2-5x+6=0, \quad \rightarrow (x-2)(x-3)=0$
- Dikkat edilecek olursa $(x-2)$ değeri orijinal rekürans ta sol tarafı, $x-3$ ise sağ taraftaki polinomu ifade etmekte. Buna göre karakteristik denklemin basit genel formunu aşağıdaki şekilde ifade edebiliriz:
- $(c_0 x^k + c_1 x^{k-1} + c_2 x^{k-2} + \dots + c_k)(x-b)^{d+1} = 0,$
- burada d , $p(n)$ polinomunun derecesidir. Bu denklem elde edildikten sonra homojen durumunda olduğu gibi çözüm yapılır.

Karakteristik denklemler kullanarak yinelemeleri çözme



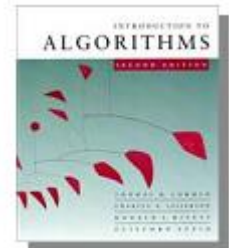
🔗 **Örnek:** Aşağıda verilen Hanoi Kulesi problemine ait reküransı çözünüz

- $t_n = 2t_{n-1} + 1; n \geq 1$ ve $t_0 = 0$;
- Burada $b=1$ $p(n)=1$ ve polinom derecesi 0 dir.
- Karakteristik denklem: $(x-2)(x-1)=0$ olur.
- $t_n = c_1 1^n + c_2 2^n, t_0 = c_1 + c_2 = 0, t_1 = 2t_0 + 1 = 1$ ise
- $t_1 = c_1 1 + 2c_2 = 1$ olur. $c_1 = -1, c_2 = 1$ bulunur
- $t_n = 2^n - 1$ elde ederiz. Sonuç olarak ;
- $t_n \in \theta(2^n)$ olur.



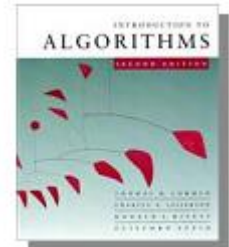
Karakteristik denklemler kullanarak yinelemeleri çözme

- $c_0 t_n + c_1 t_{n-1} + \dots + c_k t_{n-k} = b^n p(n)$ homojen olmayan denklemler için verilen basit genel formu daha da genelleştirirsek
- $c_0 t_n + c_1 t_{n-1} + \dots + c_k t_{n-k} = b_1^n p_1(n) + b_2^n p_2(n) + \dots$ formunu elde ederiz. Buna göre karakteristik denklem:
- $(c_0 x^k + c_1 x^{k-1} + c_2 x^{k-2} + \dots + c_k)(x - b_1)^{d_1+1} (x - b_2)^{d_2+1} \dots = 0,$



Karakteristik denklemler kullanarak yinelemeleri çözme

- **Örnek:** $n \geq 1$ ve $t_0=0$ başlangıç şartları için $t_n = 2t_{n-1} + n + 2^n$ problemine ait reküransı çözünüz
- $t_n - 2t_{n-1} = n + 2^n$, burada $b_1=1$, $p_1(n)=n$, $b_2=2$, $p_2(n)=1$, $d_1=1$, $d_2=0$, n polinom derecesidir.
- Karakteristik denklem: $(x-2)(x-1)^2(x-2)=0$ olur. Kökler, 1, 1, 2, 2 dir.
- Buna göre genel çözüm $t_n = c_1 1^n + c_2 n 1^n + c_3 2^n + c_4 n 2^n$
- $t_1 = 0 + 1 + 2^1 = 3, t_2 = 12, t_3 = 35$
- $n = 0$ için $c_1 + c_3 = 0$,
- $n = 1$ için $c_1 + c_2 + 2c_3 + 2c_4 = 3$,
- $n = 2$ için $c_1 + 2c_2 + 4c_3 + 8c_4 = 12$,
- $n = 3$ için $c_1 + 3c_2 + 8c_3 + 24c_4 = 35$,
- $t_n = -2 - n + 2^{n+1} + n2^n$ elde ederiz. Sonuç olarak ;
- $t_n \in \theta(n2^n)$ olur.

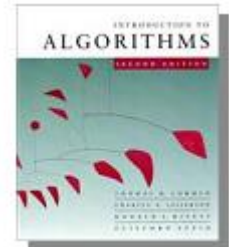


Karakteristik denklemler kullanarak yinelemeleri çözme

- Çözüm yolu: Gaus yok etme yöntemi , bilinmeyenlerin ileriye doğru elenmesi. İlk adım ilk bilinmeyi (c_1), 2. denklemden n. Denkleme kadar elemektir.
- 2.denklem
- $a_{21}-(a_{21}/a_{11})*a_{11} + a_{22}-(a_{21}/a_{11})*a_{12} +..+ a_{2n}-(a_{21}/a_{11})*a_{1n} = c_2-(a_{21}/a_{11})*c_1$
- 3. denklem
- $a_{31}-(a_{31}/a_{11})*a_{11} + a_{32}-(a_{31}/a_{11})*a_{12} +..+ a_{3n}-(a_{31}/a_{11})*a_{1n} = c_3-(a_{31}/a_{11})*c_1$
- n.denklem
- $a_{n1}-(a_{n1}/a_{11})*a_{11} + a_{n2}-(a_{n1}/a_{11})*a_{12} +..+ a_{nn}-(a_{n1}/a_{11})*a_{1n} = c_n-(a_{n1}/a_{11})*c_1$
- Buna göre ilk durumda matrisimiz

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 2 & 2 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 8 & 24 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 12 \\ 35 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 2 \\ 0 & 2 & 3 & 8 \\ 0 & 3 & 7 & 24 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 12 \\ 35 \end{bmatrix}$$

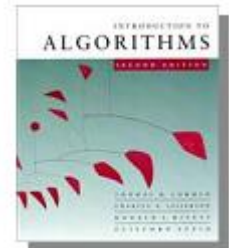
Karakteristik denklemler kullanarak yinelemeleri çözme



- İkinci adım ikinci bilinmeyeni (c_2), 3. denklemden n. denkleme kadar elemektir.
- 3.denklem
- $a_{32} - (a_{32}/a_{22}) * a_{22} + a_{33} - (a_{32}/a_{22}) * a_{23} + \dots + a_{3n} - (a_{32}/a_{22}) * a_{2n} = c_3 - (a_{32}/a_{22}) * c_2$
- n.denklem
- $a_{n2} - (a_{n2}/a_{22}) * a_{22} + a_{n3} - (a_{n2}/a_{22}) * a_{23} + \dots + a_{nn} - (a_{n2}/a_{22}) * a_{2n} = c_n - (a_{n2}/a_{22}) * c_2$

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 2 \\ 0 & 2 & 3 & 8 \\ 0 & 3 & 7 & 24 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 12 \\ 35 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 4 & 18 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 6 \\ 26 \end{bmatrix}$$

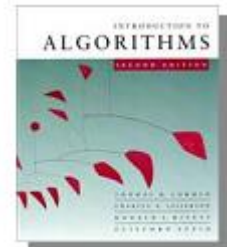
Karakteristik denklemler kullanarak yinelemeleri çözme



- Diğer adımlarda benzer şekilde yapılır.

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 4 & 18 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 6 \\ 26 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 6 \\ 2 \end{bmatrix}$$

- $c_4 = 1, c_3 = 2, c_2 = -1, c_1 = -2$ olur.
- $T(n) = -2 - n + 2^{n+2} + n2^n$
- $T(n) \in \theta(n2^n)$

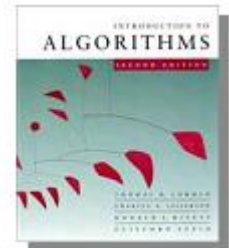


Karakteristik denklemler kullanarak yinelemeleri çözme

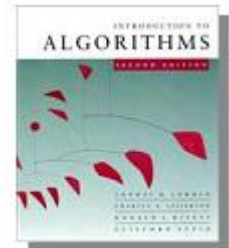
Değişkenlerin Değişimi:

- $T(n)$ şeklinde verilen bir yinelemeyi değişkenlerin değişimi ile t_k şeklinde yeni bir yineleme yazılabilir.
- **Örnek:** 2 nin kuvveti şeklinde verilen n için aşağıda verilen yinelemeyi çözünüz.
- $T(n) = 4T(n/2) + n, n > 1$
- n , değerini 2^k (burada $k = \log n$ dir) ile yer değiştirirsek $T(2^k) = 4T(2^{k-1}) + 2^k$, elde ederiz.
- Eğer $t_k = T(2^k) = T(n)$ ise bunu , $t_k = 4t_{k-1} + 2^k$ şeklinde yazabiliriz. Yeni yinelemeyi çözersek $(x-4)(x-2)=0$ karakteristik denklemini elde ederiz.
- $t_k = c_1 4^k + c_2 2^k$
- k yerine n değerini yazarsak,
- $T(n) = c_1 n^2 + c_2 n$ buluruz. $T(n) \in O(n^2)$

Karakteristik denklemler kullanarak yinelemeleri çözme



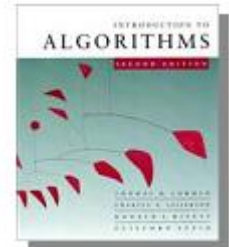
- **Örnek:** 2 nin kuvveti şeklinde verilen n için aşağıda verilen yinelemeyi çözünüz. $T(n) = 2T(n/2) + n \log n, n > 1$
- n , değerini 2^k (burada $k = \log n$ dir) ile yer değiştirirsek $T(2^k) = 2T(2^{k-1}) + k2^k$, elde ederiz.
- Eğer $t_k = T(2^k) = T(n)$ ise bunu, $t_k = 2t_{k-1} + k2^k$ şeklinde yazabiliriz. Yeni yinelemeyi çözersek: $(t_k - 2t_{k-1}) = k2^k$, burada $b=2$, $p(k)=k$ ve $d=1$ olduğundan)
- $(x-2)^3=0$ karakteristik denklemini elde ederiz ve
- $t_k = c_1 2^k + c_2 k 2^k + c_3 k^2 2^k$
- k yerine n değerini yazarsak,
- $T(n) = c_1 n + c_2 n \log n + c_3 n \log^2 n$ buluruz.
- $T(n) \in O(n \log^2 n)$



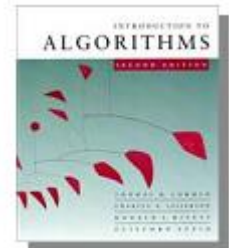
Karakteristik denklemler kullanarak yinelemeleri çözme

- **Örnek:** 2 nin kuvveti şeklinde verilen n için aşağıda verilen yinelemeyi çözünüz. $T(n) = 3T(n/2) + cn$ (c bir sabittir ve $n = 2^k > 1$)
- n , değerini 2^k (burada $k = \log n$ ' dir) ile yer değiştirirsek $T(2^k) = 3T(2^{k-1}) + c2^k$, elde ederiz.
- Eğer $t_k = T(2^k) = T(n)$ ise bunu, $t_k = 3t_{k-1} + c2^k$ şeklinde yazabiliriz. Yeni yinelemeyi çözersek: $(t_k - 3t_{k-1}) = c2^k$, burada $b=2$, $p(k)=c$ ve $d=0$ olduğundan $(x-3)(x-2)=0$ karakteristik denklemini elde ederiz ve
- $t_k = c_1 3^k + c_2 2^k$
- k yerine n değerini yazarsak,
- $T(n) = c_1 3^{\log n} + c_2 n$, ($a^{\log b} = b^{\log a}$ olduğundan)
- $T(n) = c_1 n^{\log 3} + c_2 n$ buluruz.
- $T(n) \in O(n^{\log 3})$

Karakteristik denklemler kullanarak yinelemeleri çözme

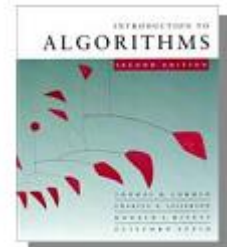


- **Aralık dönüşümleri (Range Transformations):** Yinelemelerin çözümünde değişkenlerin değişimi yerine bazen aralık dönüşümü kullanmak daha faydalı olabilir.
- **Örnek:** $T(n) = nT(n/2)^2$, $n > 1$, $T(1) = 6$
- n , değerini 2^k (burada $k = \log n$ dir) ile yer değiştirirsek $T(2^k) = 2^k T(2^{k-1})^2$, elde ederiz.
- $t_k = T(2^k) = T(n)$ ise, $t_k = 2^k t_{k-1}^2$, $k > 0$ için $t_0 = 6$
- İlk bakışta gördüğümüz tekniklerin hiç biri bu yineleme için uygulanamaz çünkü hem doğrusal değil, hem de katsayılarından biri sabit değildir.
- Aralık dönüşümü yapmak için $V_k = \log t_k$ koyarak yeni bir yineleme oluşturulur.



Karakteristik denklemler kullanarak yinelemeleri çözme

- ⑥ $V_k = k + 2V_{k-1}$, $k > 0$ için başlangıç şartları $V_0 = \log 6 = \log 2 * 3 = 1 + \log 3$
- $V_k = k + 2V_{k-1}$ için karakteristik denklem $(x-2)(x-1)^2 = 0$ ve
- $V_k = c_1 2^k + c_2 1^k + c_3 k 1^k$
- $V_k = k + 2V_{k-1}$ denkleminde $V_0 = 1 + \log 3$, $V_1 = 3 + 2\log 3$, $V_2 = 8 + 4\log 3$ bulunur ve $V_k = c_1 2^k + c_2 1^k + c_3 k 1^k$
- $V_0 = 1 + \log 3 = c_1 + c_2$
- $V_1 = 3 + 2\log 3 = 2c_1 + c_2 + c_3$
- $V_2 = 8 + 4\log 3 = 4c_1 + c_2 + 4c_3$
- $c_1 = 3 + \log 3, c_2 = -2, c_3 = -1$
- $V_k = (3 + \log 3)2^k - k - 2$



Karakteristik denklemler kullanarak yinelemeleri çözme

• Sonunda $t_k = 2^{V_k}$

• $2^{V_k} = 2^{(3+\log 3) \cdot 2^k - k - 2} \rightarrow t_k = 2^{(3+\log 3) \cdot 2^k - k - 2}$

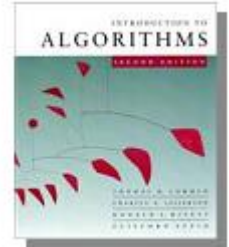
• k yerine n değerini yazarsak,

• $T(n) = 2^{(3+\log 3) \cdot n - \log n - 2} \rightarrow T(n) = 2^{3n + n \log 3} / (2^{\log n} * 2^2)$

• $T(n) = 2^{3n-2} * \frac{3^{n \log 2}}{n} = 2^{3n-2} * \frac{3^n}{n}$

• $T(n) = (2^{3n-2} 3^n) / n$

• $T(n) \in O(2^{3n} 3^n)$



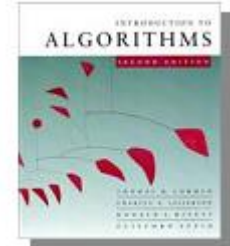
Böl-ve-Fethet (Divide & Conquer)

- Böl ve fethet tekniğiyle algoritma tasarımı:
 - Problem kendisine benzer küçük boyutlu alt problemlere bölünür. Alt problemler çözülür ve bulunan çözümler birleştirilir.
 - **Divide:** Problem iki veya daha fazla alt problemlere bölünür.
 - **Conquer:** Alt problemleri özyinelemeli olarak çözüp, onları **fethet**.
 - **Combine:** Alt problem çözümlerini **birleştirir**.

Merge Sort (Birleştirme sıralaması) Algoritması

- **1. Böl:** Eğer S en az iki elemana sahipse (S sıfır veya bir elemana sahipse hiçbir işlem yapılmaz), bütün elemanlar S 'e n alınır ve S_1 ve S_2 adlı iki alana yerleştirilir, her biri S dizisinin yarısına sahiptir, (örn. S_1 ilk $\lceil n/2 \rceil$ elemana ve S_2 ise ikinci $\lfloor n/2 \rfloor$ elemana sahiptir).
- **2. Fethet:** S_1 ve S_2 Merge Sort kullanılarak sıralanır.
- **3. Birleştir:** S_1 and S_2 içindeki sıralı elemanlar tekrar S içerisine tek bir sıralı dizi oluşturacak şekilde aktarılır.

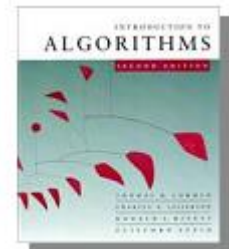
Birleştirme sıralaması



- 1. Bölmek: Kolay.
- 2. Hükm etmek: 2 alt dizilimi özyinelemeli sıralama.
- 3. Birleştirmek: Doğrusal-zamanda birleştirme.

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

altproblem sayısı *altproblem boyutu* *bölme ve birleştirme işi*



Master teoremi (hatırlatma)

$$T(n) = a T(n/b) + f(n)$$

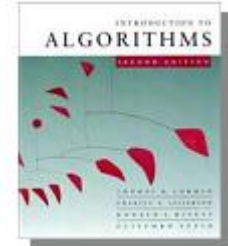
DURUM 1: $f(n) = O(n^{\log_b a - \varepsilon})$, sabit $\varepsilon > 0$
 $\Rightarrow T(n) = \Theta(n^{\log_b a})$.

DURUM 2: $f(n) = \Theta(n^{\log_b a} \lg^k n)$, sabit $k \geq 0$
 $\Rightarrow T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$.

DURUM 3: $f(n) = \Omega(n^{\log_b a + \varepsilon})$, sabit $\varepsilon > 0$,
 ve düzenleyici koşul (regularity condition).
 $\Rightarrow T(n) = \Theta(f(n))$.

Birleştirme sıralaması: $a = 2, b = 2 \Rightarrow n^{\log_b a} = n^{\log_2 2} = n$
 \Rightarrow **DURUM 2** ($k = 0$) $\Rightarrow T(n) = \Theta(n \lg n)$.

İkili arama (Binary Search)



○ Sıralı dizilimin bir elemanını bulma:

INPUT: $A[1..n]$ – sıralı (azalmayan) integer sayı dizisi, s – aranan integer sayı.

OUTPUT: j bulunan sayının indeksi $A[j] = s$. *NIL*, if $\forall j (1 \leq j \leq n): A[j] \neq s$

Binary-search(A, p, r, s):

```

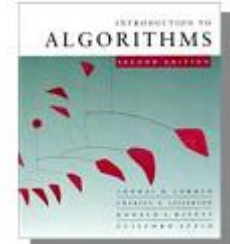
    if  $p = r$  then
        if  $A[p] = s$  then return  $p$ 
        else return NIL
     $q \leftarrow \lfloor (p+r)/2 \rfloor$ 
    ret  $\leftarrow$  Binary-search( $A, p, q, s$ )
    if ret = NIL then
        return Binary-search( $A, q+1, r, s$ )
    else return ret

```

- 1. **Böl**: Orta elemanı belirle.
- 2. **Hükmet**: 1 alt dizilimde özyinelemeli arama yap.
- 3. **Birleştir**: Kolay.
- Örnek: 9' u bul.

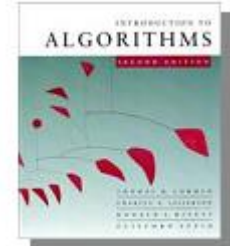
3 5 7 8 9 12 15

İkili arama (Binary Search)



3	5	7	8	9	12	15
3	5	7	8	9	12	15
3	5	7	8	9	12	15
3	5	7	8	9	12	15
3	5	7	8	9	12	15

İkili arama için yineleme



$$T(n) = 1 T(n/2) + \Theta(1)$$

altproblem sayısı

altproblem boyutu

*bölme ve
birleştirme işi*

$$n^{\log_b a} = n^{\log_2 1} = n^0 = 1 \Rightarrow \text{DURUM 2 } (k = 0) \\ \Rightarrow T(n) = \Theta(\lg n) .$$

Bir sayının üstellenmesi



- **Problem:** a^n 'yi $n \in \mathbb{N}$ iken hesaplama.
- **Saf (Naive) algorithm:** $\Theta(n)$.
- **Böl-ve-fethet algoritması:**

Algorithm Power(x, n):

Input: x sayısı ve n tamsayısı, $n \geq 0$

Output: x^n değeri

if $n = 0$ **then**

return 1

if n is odd **then**

$y = \text{Power}(x, (n - 1) / 2)$

return $x \cdot y \cdot y$

else

$y = \text{Power}(x, n / 2)$

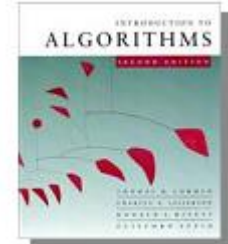
return $y \cdot y$

Her özyineli çağırmda n sayısını 2'ye bölüyoruz; dolayısıyla, $\log n$ özyineli çağrı yaparız. Bu metod $O(\log n)$ zamanına sahiptir.

Burada ara sonucu y değişkeni ile göstermemiz önemli; şayet metod çağırma yazarsak metod 2 defa çağırılmış olur.

$$a^n = \begin{cases} a^{n/2} \cdot a^{n/2} & n \text{ çift sayıysa;} \\ a^{(n-1)/2} \cdot a^{(n-1)/2} \cdot a & n \text{ tek sayıysa.} \end{cases}$$

$$T(n) = T(n/2) + \Theta(1) \Rightarrow T(n) = \Theta(\lg n) .$$



Fibonacci sayıları

Özyinelemeli tanım:

$$F_n = \begin{cases} 0 & \text{eğer } n = 0; \\ 1 & \text{eğer } n = 1; \\ F_{n-1} + F_{n-2} & \text{eğer } n \geq 2 \text{ ise.} \end{cases}$$

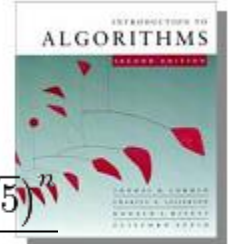
0 1 1 2 3 5 8 13 21 34 ...

Saf özyinelemeli algoritma: $\Omega(\phi^n)$
 (üstel zaman), buradaki $\phi = (1 + \sqrt{5})/2$
altın oran'dır (*golden ratio*).

$$\varphi = \frac{1 + \sqrt{5}}{2} \approx 1.6180339887 \dots$$

$$\psi = \frac{1 - \sqrt{5}}{2} = 1 - \varphi = -\frac{1}{\varphi} \approx -0.6180339887$$

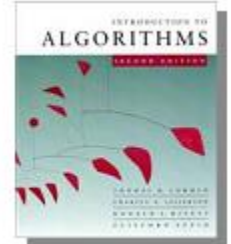
Fibonacci sayılarını hesaplama



$$F_n = F(n) = \begin{cases} 0 & n = 0; \\ 1 & n = 1; \\ F(n-1) + F(n-2) & n > 1. \end{cases} = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}} = \frac{\varphi^n - (\varphi - \sqrt{5})^n}{\sqrt{5}}$$

○ Aşağıdan yukarıya algortiması:

- $F_0, F_1, F_2, \dots, F_n$ 'i sırayla, her sayı iki öncekinin toplamı olacak şekilde hesaplayın.
- Yürütüm süresi: $\Theta(n)$.
- **Saf özyinelemeli kare alma (Naive recursive squaring) algortiması:**
 - $F_n = \varphi^n / \sqrt{5}$ yakın tamsayı yuvarlaması.
 - **Özyinelemeli kare alma algortiması:** $\Theta(\lg n)$ zamanı.
 - Bu yöntem güvenilir değildir, çünkü yüzer-nokta aritmetiği yuvarlama hatalarına gebedir.



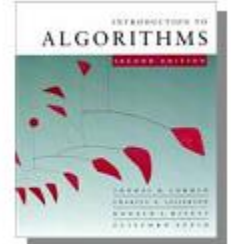
Özyineleme ile kare alma (Recursive squaring)

Teorem:
$$\begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n.$$

Algoritma: Özyineleme ile kare alma.
Süre = $\Theta(\lg n)$.

Teoremin ispatı. (n 'de tümevarım)

Taban ($n = 1$):
$$\begin{bmatrix} F_2 & F_1 \\ F_1 & F_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^1$$

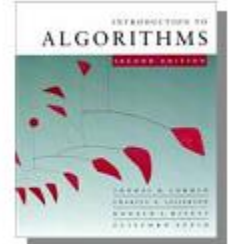


Özyineleme ile kare alma (Recursive squaring)

Tümevarım adımı ($n \geq 2$):

$$\begin{aligned}
 \begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix} &= \begin{bmatrix} F_n & F_{n-1} \\ F_{n-1} & F_{n-2} \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{n-1} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n
 \end{aligned}$$



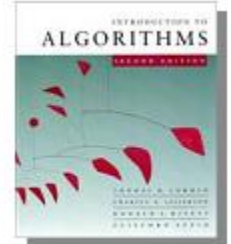


Matrislerde çarpma

$$\left. \begin{array}{l} \text{Girdi:} \quad A = [a_{ij}], B = [b_{ij}]. \\ \text{Çıktı:} \quad C = [c_{ij}] = A \cdot B. \end{array} \right\} i, j = 1, 2, \dots, n.$$

$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix}$$

$$c_{ij} = \sum_{k=1}^n a_{ik} \cdot b_{kj}$$

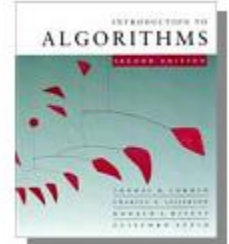


Matrislerde çarpma

Standart algoritma

```
for  $i \leftarrow 1$  to  $n$            (i 1'den n'ye kadar)  
  do for  $j \leftarrow 1$  to  $n$    (j 1'den n'ye kadar)  
    do  $c_{ij} \leftarrow 0$   
      for  $k \leftarrow 1$  to  $n$   
        do  $c_{ij} \leftarrow c_{ij} + a_{ik} \cdot b_{kj}$ 
```

Koşma süresi = $\Theta(n^3)$



Böl-ve-fethet algoritması

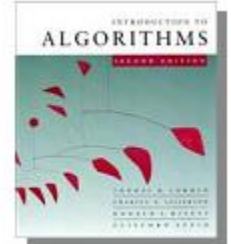
FİKİR:

$n \times n$ matris = $(n/2) \times (n/2)$ altmatrisin 2×2 matrisi:

$$\begin{bmatrix} r & s \\ t & u \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

$$C = A \cdot B$$

$\left. \begin{array}{l} r = ae + bg \\ s = af + bh \\ t = ce + dh \\ u = cf + dg \end{array} \right\} \begin{array}{l} \text{recursive (özyinelemeli)} \\ 8 \text{ çarpma } (n/2) \times (n/2) \text{ altmatriste,} \\ 4 \text{ toplama } (n/2) \times (n/2) \text{ altmatriste.} \end{array}$



Böl-ve-Fethet algoritmasının çözümlemesi

$$T(n) = 8T(n/2) + \Theta(n^2)$$

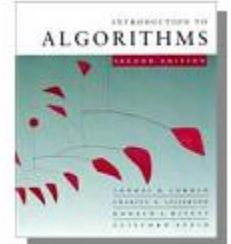
altmatris sayısı

altmatris boyutu

*altmatrisleri
toplama işi*

$$n^{\log_b a} = n^{\log_2 8} = n^3 \Rightarrow \text{DURUM 1} \Rightarrow T(n) = \Theta(n^3)$$

Sıradan algoritmadan daha iyi değil.



Strassen'in fikri

- 2×2 matrisleri yalnız 7 özyinelemeli çarpmayla çöz.

$$P_1 = a \cdot (f - h)$$

$$P_2 = (a + b) \cdot h$$

$$P_3 = (c + d) \cdot e$$

$$P_4 = d \cdot (g - e)$$

$$P_5 = (a + d) \cdot (e + h)$$

$$P_6 = (b - d) \cdot (g + h)$$

$$P_7 = (a - c) \cdot (e + f)$$

$$r = P_5 + P_4 - P_2 + P_6$$

$$s = P_1 + P_2$$

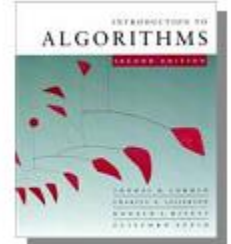
$$t = P_3 + P_4$$

$$u = P_5 + P_1 - P_3 - P_7$$

7 çarp., 18 topl. / çıkar.

Not: Çarpma işleminde sırabagimsizlik yok!

Strassen'in fikri



- **2×2** matrisleri yalnız **7** özyinelemeli çarpmayla çöz.

$$P_1 = a \cdot (f - h)$$

$$P_2 = (a + b) \cdot h$$

$$P_3 = (c + d) \cdot e$$

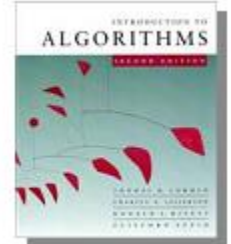
$$P_4 = d \cdot (g - e)$$

$$P_5 = (a + d) \cdot (e + h)$$

$$P_6 = (b - d) \cdot (g + h)$$

$$P_7 = (a - c) \cdot (e + f)$$

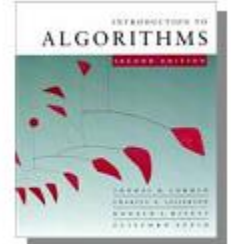
$$\begin{aligned} r &= P_5 + P_4 - P_2 + P_6 \\ &= (a + d)(e + h) \\ &\quad + d(g - e) - (a + b)h \\ &\quad + (b - d)(g + h) \\ &= ae + ah + de + dh \\ &\quad + dg - de - ah - bh \\ &\quad + bg + bh - dg - dh \\ &= ae + bg \end{aligned}$$



Strassen'in algoritması

- **1. Böl:** A ve B 'yi $(n/2) \times (n/2)$ altmatrislere böl. $+$ ve $-$ kullanarak çarpılabilecek terimler oluştur. ($\Theta(n^2)$)
- **2. Fethet:** $(n/2) \times (n/2)$ altmatrislerde özyinelemeli **7** çarpma yap ($P_1, P_2, P_3, \dots P_7$.)
- **3. Birleştir:** $+$ ve $-$ kullanarak $(n/2) \times (n/2)$ altmatrislerde C 'yi oluştur. ($\Theta(n^2)$)

$$T(n) = 7 T(n/2) + \Theta(n^2)$$



Strassen'in algoritması

$$T(n) = 7 T(n/2) + \Theta(n^2)$$

$$n^{\log_b a} = n^{\log_2 7} \approx n^{2.81} \Rightarrow \text{DURUM 1} \Rightarrow T(n) = \Theta(n^{\lg 7})$$

- **2.81** değeri **3**' den çok küçük görünmeyebilir ama, fark üstelde olduğu için, yürütüm süresine etkisi kayda değerdir. Aslında, **$n \geq 32$** değerlerinde Strassen'in algoritması günün makinelerinde normal algoritmadan daha hızlı çalışır. **Bugünün en iyi değeri** (teorik merak açısından): **$\Theta(n^{2.376...})$**

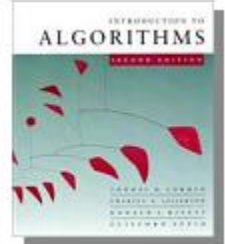
Böl ve Fethet

VLSI (Very Large Scale Integration) yerleşimi (Çok Büyük Çapta Tümleşim)

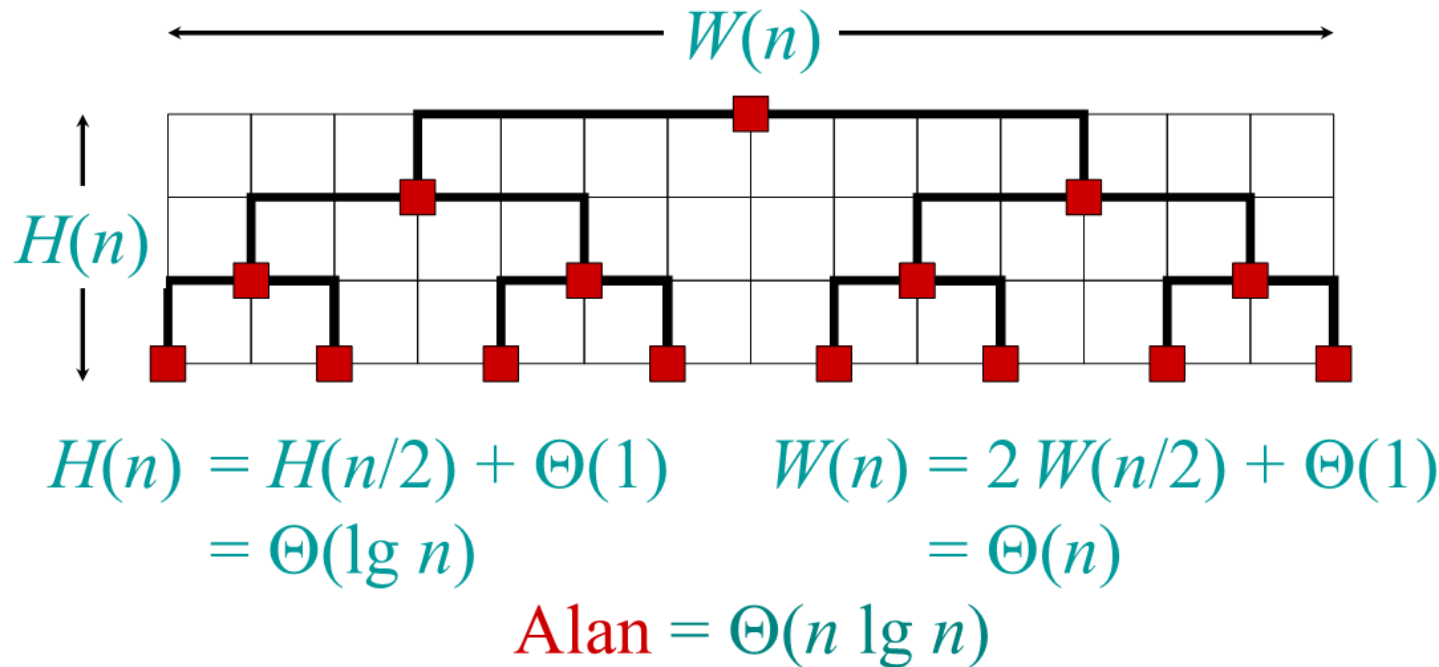
- Bilgisayar çipleri yada yongaları bildiğiniz gibi çok büyük çapta tümleşim kullanırlar.
- Elimizde bir devre olduğunu düşünelim ve bu devrenin de bir ikili ağaç olduğunu kabul edelim. Ama şimdilik bu devrenin bir kısmını ele alalım ama siz bunu tüm devre kabul edin.
- **Problem:** n yaprağı olan tam bir ikili ağacı en az alan kullanarak bir ızgaraya gömmek.



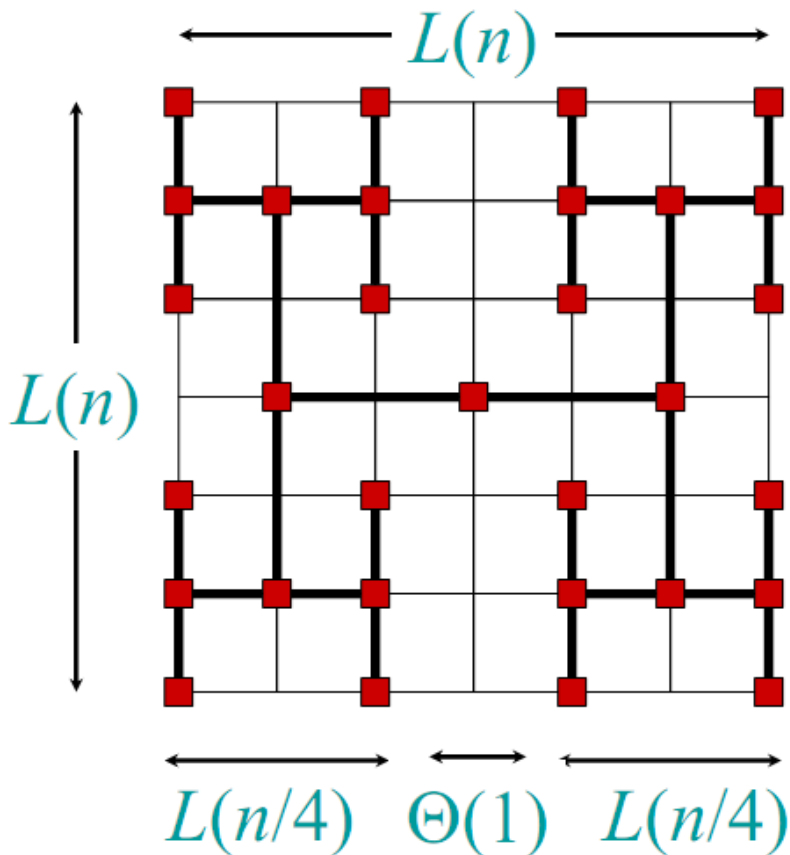
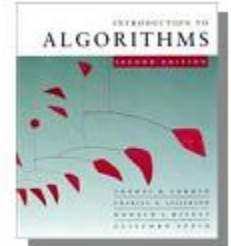
VLSI (Very Large Scale Integration) yerleşimi (Çok Büyük Çapta Tümlleşim)



- **Problem:** n yaprağı olan tam bir ikili ağacı en az alan kullanarak bir ızgaraya gömmek.



H-ağacını gömme



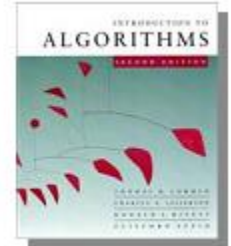
$$L(n) = 2L(n/4) + \Theta(1)$$

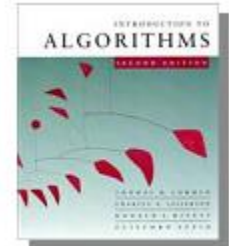
$$= \Theta(\sqrt{n})$$

$$\text{Alan} = \Theta(n)$$

Sonuç

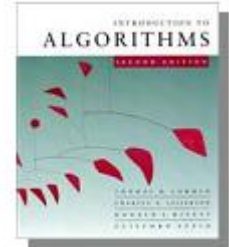
- Böl ve Fethet algoritma tasarımının güçlü tekniklerinden sadece biridir.
- Böl ve Fethet algoritmaları yinelemeler ve Ana (Master) metot kullanarak çözümlenebilir.
- Böl ve Fethet stratejisi genellikle verimli algoritmalara götürür.





Ortak Reküranslar

Rekürans İlişkisi	Kapalı Form	Örnek
$c(1) = a$		
$c(n) = b + c(n-1)$	$c(n) = O(n)$	Linear search
$c(n) = b*n + c(n-1)$	$c(n) = O(n^2)$	Quicksort
$c(n) = b + c(n/2)$	$c(n) = O(\log(n))$	Binary search
$c(n) = b*n + c(n/2)$	$c(n) = O(n)$	
$c(n) = b + kc(n/k)$	$c(n) = O(n)$	
$c(n) = b*n + 2c(n/2)$	$c(n) = O(n \log(n))$	Mergesort
$c(n) = b*n + kc(n/k)$	$c(n) = O(n \log(n))$	
$c(2) = b$ $c(n) = c(n-1) + c(n-2) + d$	$c(n) = O(2^n)$	Fibonacci



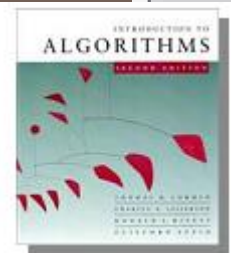
Sorular

- 1. $T(n) = 3T(\sqrt{2n}) + 2$ tekrarlı bağıntısını çözünüz.
- 2. $T(n) = 3T(\lfloor n/5 \rfloor) + n$ tekrarlı bağıntısının çözümünü iteratif yolla gerçekleştiriniz. Bu bağıntının Özyineleme ağacı nedir?
- 3. Özyineleme ağacını kullanarak $T(n) = T(n/3) + T(2n/3) + n$ bağıntısının çözümünü elde ediniz.
- 4. $b \geq 1$ bir sabit olmak üzere $T(n) = T(n/b) + T(b) + n$ tekrarlı bağıntısının Özyineleme ağacını elde ediniz ve bu bağıntının çözümü nedir?
- 5. $0 < a < 1$ sabit olmak üzere $T(n) = T(an) + T((1-a)n) + n$ tekrarlı bağıntısının Özdevinim ağacını elde ediniz ve asimptotik davranışı hakkında bilgi veriniz.



Sorular

- 6.
 - a) $T(n) = \sqrt{n}T(\sqrt{n}) + n$ çalışma zamanı mertebesinin $O(n \log \log n)$ olduğunu iterasyon veya öz yineleme ağacı ile bulunuz.
 - b) $T(n) = \sqrt{n}T(n/2) + n$
- 7. Master yöntemini kullanarak aşağıdaki tekrarlı bağıntıları çözünüz.
 - a) $T(n) = 3T(n/3) + n$
 - b) $T(n) = 3T(n/3) + n^2$
 - c) $T(n) = 3T(n/3) + n^3$
 - d) $T(n) = 3T(n/3) + n^k$
- 8. Aşağıdaki tekrarlı bağıntı verilmiş olsun.
 - $T(n) = 2T(n/3) + \lg(n)$
 - a) İteratif yöntem ile bu bağıntının mertebesini (çalışma zamanını) elde ediniz.
 - b) Master yöntemi ile bu bağıntının mertebesini (çalışma zamanını) elde ediniz.



Sorular

- 9. Aşağıdaki tekrarlı bağıntıları karakteristik denklem ve üreten fonksiyon yöntemleri ile çözünüz.
 - a) $a_n = 5a_{n-1} - 6a_{n-2}$, $a_1 = 36$ ve $a_0 = 0$
 - b) $a_n = 3a_{n-1} - 2a_{n-2} + 2^{n-1} + 2 \cdot 3^n$, $a_1 = 29$ ve $a_0 = 9$
 - c) $a_n = a_{n-2} + 4n$, $a_1 = 4$ ve $a_0 = 1$
 - d) $a_n = 3a_{n-1} - 2a_{n-2} + 3 \cdot 2^{2n-1}$, $a_1 = 12$ ve $a_0 = 0$
- 10. Master teoremini kullanarak aşağıdaki bağıntının mertebesini (çalışma zamanını) elde ediniz.
 - $T(n) = 16T(n/4) + O(n^2)$
- 11. $f(n) = n^2 + 4n \log n + 900$ ve $g(n) = n^2 + 45 \log n + h(n)$ fonksiyonları verilmiştir ve $h(n)$ lineer olan bir polinomdur. $f(n)$ ile $g(n)$ arasındaki asimptotik ilişki nedir? $f(n)$ ve $g(n)$ arasındaki asimptotik ilişkiyi belirlerken $h(n)$ polinomuna ihtiyaç var mıdır? Hangi durumlarda ihtiyaç duyulur veya duyulmaz?

4.Hafta

Sıralama Algoritmaları
Çabuk Sıralama, Rastgele
Algoritmalar