

Tümevarım ve Özyineleme

CSC-2259 Ayrık Yapılar

Konstantin Busch - LSU

1

Tümevarım

Tümevarım çok kullanışlı bir ispat tekniğidir.

Bilgisayar bilimlerinde, tümevarım algoritmalarının özelliklerini kanıtlamak için kullanılır.

Tümevarım ve öz yineleme bir biri ile ilişkilidir.

- Özyineleme algoritmalar için bir tanımlama metotudur
- Tümevarım öz yineleme algoritmaları için uygun bir ispat metodudur.

Konstantin Busch - LSU

2

$P(n)$ önermesinin doğru olduğunu ispat etmek için tümevarım yöntemini kullanınız:

Temel tümevarım: $P(1)$ doğru olduğunu ispatlayınız

Tümevarım hipotezi: $P(k)$ doğru olduğunu varsayınız.
(her hangi bir k pozitif tam sayısı için)

Tümevarım basamğı:

$P(k + 1)$ doğru olduğunu varsayınız.

Konstantin Busch - LSU

3

Tümevarım hipotezi: $P(k)$ doğru olduğunu varsayınız.

(her hangi bir k pozitif tam sayısı için)

Tümevarım basamğı: $P(k + 1)$
doğru olduğunu varsayınız.

Tümevarım basamaklarını diğer
bir ifade ile ispatlarsak:

$$P(k) \rightarrow P(k + 1)$$

k her pozitif tam sayı için

Konstantin Busch - LSU

4

Temel tümevarım

$$P(1)$$

Doğru

Tümevarım basamağı

$$P(k) \rightarrow P(k+1)$$

Doğru

Bütün pozitif tamsayılar için önerme doğrudur

$$P(1) \rightarrow P(2) \rightarrow P(3) \rightarrow P(4) \rightarrow \dots$$

Çıkarım kuralı olarak tümevarım:

$$[P(1) \wedge \forall k(P(k) \rightarrow P(k+1))] \rightarrow \forall n P(n)$$

Teorem: $P(n): 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$

İspat:

Temel tümevarım: $P(1): 1 = \frac{1(1+1)}{2}$

Tümevarım hipotezi: olduğunu varsayalım

$$P(k): 1 + 2 + \dots + k = \frac{k(k+1)}{2}$$

Tümevarım basamğı: İspatlarsak

$$P(k+1): 1 + 2 + \dots + k + (k+1) = \frac{(k+1)((k+1)+1)}{2}$$

K. Busch - LSU

7

Tümevarım basamğı:

$$\begin{aligned} P(k+1): & 1 + 2 + \dots + k + (k+1) && \text{(tümevarım hipotezi)} \\ &= \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k(k+1) + 2(k+1)}{2} \\ &= \frac{(k+1)((k+1)+1)}{2} \end{aligned}$$

İspat sonu

Konstantin Busch - LSU

8

Harmonik sayılar

$$H_j = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{j}$$

$$j = 1, 2, 3, \dots$$

Örnek: $H_4 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = \frac{25}{12}$

Teorem: $H_{2^n} \geq 1 + \frac{n}{2} \quad n \geq 0$

İspat:

Temel tümevarım: $n = 0$

$$H_{2^n} = H_{2^0} = H_1 = 1 = 1 + \frac{0}{2} = 1 + \frac{n}{2}$$

Tümevarım hipotezi: $n = k$

Beklendiğini varsayalım: $H_{2^k} \geq 1 + \frac{k}{2}$

Tümevarım basamağı: $n = k + 1$

Göstereceğiz: $H_{2^{k+1}} \geq 1 + \frac{k+1}{2}$

Konstantin Busch - LSU

11

$$H_{2^{k+1}} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{2^k} + \frac{1}{2^k + 1} + \cdots + \frac{1}{2^{k+1}}$$

$$= H_{2^k} + \frac{1}{2^k + 1} + \cdots + \frac{1}{2^{k+1}}$$

$$\geq \left(1 + \frac{k}{2}\right) + \frac{1}{2^k + 1} + \cdots + \frac{1}{2^{k+1}}$$

Tümevarım
hipotezinden

$$\geq \left(1 + \frac{k}{2}\right) + 2^k \cdot \frac{1}{2^{k+1}}$$

$$= \left(1 + \frac{k}{2}\right) + \frac{1}{2}$$

$$= 1 + \frac{k+1}{2}$$

İspat sonu

Konstantin Busch - LSU

12

Teorem: $H_{2^n} \leq 1 + n \quad n \geq 0$

İspat:

Temel tümevarım: $n = 0$

$$H_{2^n} = H_{2^0} = H_1 = 1 = 1 + 0 = 1 + n$$

Tümevarım hipotezi: $n = k$

Beklendiğini varsayalım: $H_{2^k} \leq 1 + k$

Tümevarım hipotezi: $n = k + 1$

Göstereceğiz: $H_{2^{k+1}} \leq 1 + (k + 1)$

$$H_{2^{k+1}} = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{2^k} + \frac{1}{2^k + 1} + \dots + \frac{1}{2^{k+1}}$$

$$= H_{2^k} + \frac{1}{2^k + 1} + \dots + \frac{1}{2^{k+1}}$$

$$\leq (1+k) + \frac{1}{2^k + 1} + \dots + \frac{1}{2^{k+1}}$$

Tümevarım
hipotezinden

$$\leq (1+k) + 2^k \cdot \frac{1}{2^k + 1}$$

$$\leq (1+k) + 1$$

$$= 1 + (k+1)$$

İspat

Konstantin Busch - LSU

15

Gördüğümüz gibi: $1 + \frac{n}{2} \leq H_{2^n} \leq 1 + n$

Beklendiği gibi: $H_{2^{\lfloor \log k \rfloor}} \leq H_k \leq H_{2^{\lceil \log k \rceil}}$

$$1 + \frac{\lfloor \log k \rfloor}{2} \leq H_k \leq 1 + \lceil \log k \rceil$$

$$H_k = \Theta(\log k)$$

Konstantin Busch - LSU

16

Triominos

2×2 $2^2 \times 2^2$ $2^3 \times 2^3$

boşluk boşluk boşluk

17

Teorem: Her $2^n \times 2^n$, $n \geq 1$ dama tahtasında Triominos'ın kuyruğu bir kare hareket edebilir.

İspat: Temel tümevarım: $n = 1$

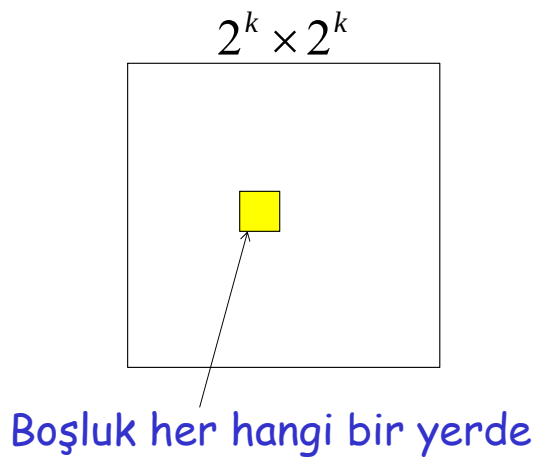
$2^1 \times 2^1$

boşluk

18

Tümevarım hipotezi: $n = k$

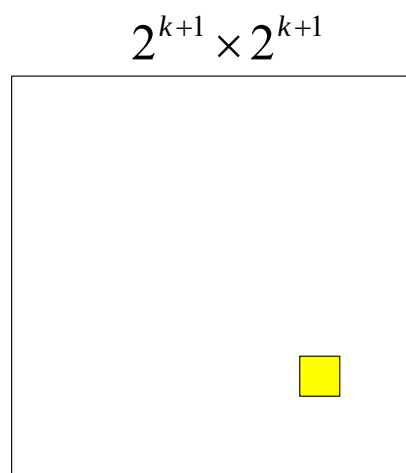
$2^k \times 2^k$ dama tahtasında boşluğun her hangi bir yere yerleştiğini varsayalım



Konstantin Busch - LSU

19

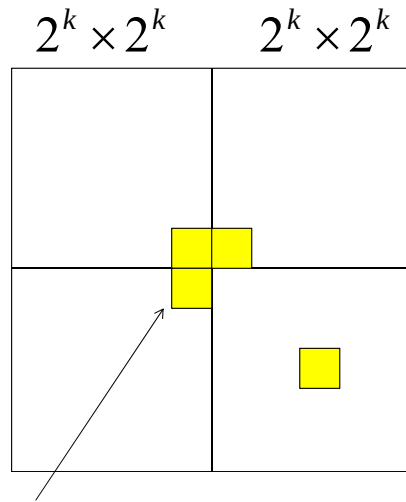
Tümevarım basamağı: $n = k + 1$



Konstantin Busch - LSU

20

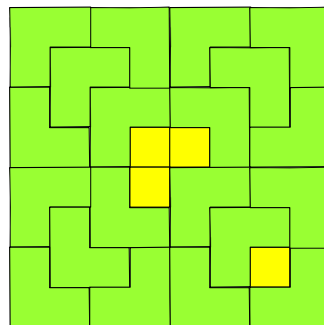
Tümevarım hipoteziyle $2^k \times 2^k$ karede boşluğun döşendiğini düşünelim.



Konstantin Busch - LSU

21

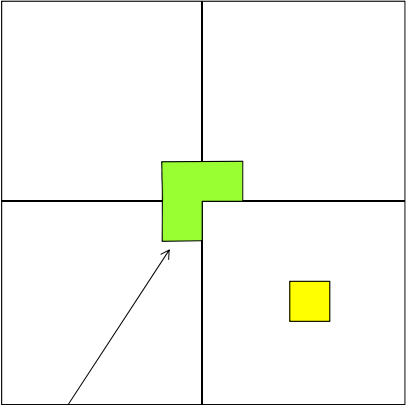
$2^3 \times 2^3$ durumda:



Konstantin Busch - LSU

22

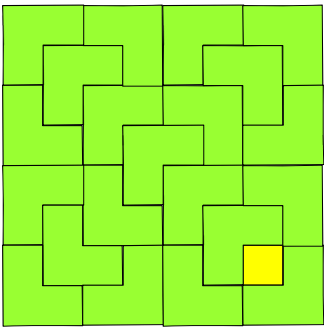
$2^k \times 2^k$ $2^k \times 2^k$



Triomine ile üç boşluğu yer değiştirelim.
Şimdi bütün alanlar döşenmiş olur.

Konstantin Busch - LSU
23

$2^3 \times 2^3$ durum için:



İspat sonu

Konstantin Busch - LSU
24

Güçlü Tümevarım

$P(n)$ Önermesi için:

Temel tümevarım: $P(1)$ doğru olduğunu ispatlayalım

Tümevarım hipotezi:

$$P(1) \wedge P(2) \wedge \cdots \wedge P(k)$$

doğru olduğunu varsayalım

Tümevarım basamağı:

$P(k+1)$ doğru olduğunu varsayalım

Konstantin Busch - LSU

25

Teorem: $n \geq 2$ her tamsayı, asal sayıların sonucudur.
(sonuçlarından en az biri asal sayıdır)

İspat: (Güçlü tümevarım)

Temel tümevarım: $n = 2$

2 sayısı bir asal sayıdır.

Tümevarım hipotezi: $2 \leq n \leq k$

2 ve k arasındaki her tamsayı asal sayıların bir sonucudur diye söylenebilir.

Konstantin Busch - LSU

26

Tümevarım basamakları: $n = k + 1$

Eğer $k + 1$ asal sayı ise ispat bitmiştir.

Eğer $k + 1$ asal sayı değilse, o bileşiktir.

$$k + 1 = a \cdot b \quad 2 \leq a, b \leq k$$

$$k + 1 = a \cdot b \quad 2 \leq a, b \leq k$$

Tümevarım hipotezine göre;

$$\left. \begin{array}{l} i, j \geq 1 \\ a = p_1 p_2 \cdots p_i \\ b = q_1 q_2 \cdots q_j \\ \text{Asal sayı} \end{array} \right\} \Rightarrow k + 1 = a \cdot b = p_1 \cdots p_i q_1 \cdots q_j$$

Asal sayı

İspat sonu

Teorem:

$n \geq 12$ için her posta gönderim bedeli
4 ve 5 sentlik posta pulları kullanılarak
ayarlanabilir.

İspat: (Güçlü tüme varım)

Tümevarım temelleri: Dört durumu inceleyelim.
(tüme varım basamakları için)

$$n = 12 = 4 + 4 + 4$$

$$n = 14 = 5 + 5 + 4$$

$$n = 13 = 4 + 4 + 5$$

$$n = 15 = 5 + 5 + 5$$

Konstantin Busch - LSU

29

Tümevarım hipotezi: $12 \leq n \leq k$

12 ile k arasındaki her posta bedelinin 4
sent ve 5 sentlik pullar kullanarak
üretilebileceğini varsayalım

$$n = a \cdot 4 + b \cdot 5$$

Tümevarım basamağı: $n = k + 1$

Eğer $12 \leq k \leq 14$ ise tümevarım basamağı
tümevarım temelinden direkt olarak
takip eder

Konstantin Busch - LSU

30

Varsayım olarak: $k \geq 15$ $k+1 = (k-3) + 4$

$$12 \leq (k-3) \leq k$$



Tümevarım hipotezi

$$(k-3) = a' \cdot 4 + b' \cdot 5$$



$$k+1 = (k-3) + 4 = (a' + 1) \cdot 4 + b' \cdot 5$$

İspat sonu

Konstantin Busch - LSU

31

Özyineleme

Özyineleme; fonksiyonların, kümelerin, algoritmaların tanımlanmasında kullanılır.

Örnek: Faktoriyel fonksiyon $f(n) = n!$

Özyineleme temel $f(0) = 1$

Özyineleme basamak: $f(n+1) = (n+1) \cdot f(n)$

Konstantin Busch - LSU

32

Faktoriyel için özyineleme algoritması

```
factorial( n ) {  
  if n = 1 then //özyineleme temel  
    return 1  
  else //özyineleme basamak  
    return n · factorial(n-1)  
}
```

Konstantin Busch - LSU

33

Fibonacci sayıları

$f_0, f_1, f_2, f_3, \dots$

Özyineleme temel: $f_0 = 0, \quad f_1 = 1$

Özyineleme basamak: $f_n = f_{n-1} + f_{n-2}$
 $n = 2, 3, 4, \dots$

Konstantin Busch - LSU

34

$$f_0 = 0$$

$$f_1 = 1$$

$$f_2 = f_1 + f_0 = 1 + 0 = 1$$

$$f_3 = f_2 + f_1 = 1 + 1 = 2$$

$$f_4 = f_3 + f_2 = 2 + 1 = 3$$

$$f_5 = f_4 + f_3 = 3 + 2 = 5$$

$$f_6 = f_5 + f_4 = 5 + 3 = 8$$

$$f_7 = f_6 + f_5 = 8 + 5 = 13$$

$\vdots$

Konstantin Busch - LSU

35

Fibonacci fonksiyonu için özyineleme algoritması

```
fibonacci( $n$ ) {  
  if  $n \in \{0,1\}$  then //özyineleme temel  
    return  $n$   
  else //özyineleme basamak  
    return fibonacci( $n-1$ ) + fibonacci( $n-2$ )  
}
```

Konstantin Busch - LSU

36

Fibonacci fonksiyonu için iterasyon algoritması

```
fibonacci( $n$ ) {  
  if  $n = 0$  then  $y \leftarrow 0$   
  else {  
     $x \leftarrow 0$   
     $y \leftarrow 1$   
    for  $i \leftarrow 1$  to  $n - 1$  do {  
       $z \leftarrow x + y$   
       $x \leftarrow y$   
       $y \leftarrow z$   
    }  
    return  $y$   
  }  
}
```

Konstantin Busch - LSU

37

Teorem: $n \geq 3$ için $f_n > \delta^{n-2}$

$$\delta = \frac{1 + \sqrt{5}}{2} \quad (\text{altın oran})$$

İspat: Tümevarım (güçlü) ile ispat

Tümevarım temel: $n = 3$ $n = 4$

$$f_3 = 2 > \delta$$

$$f_4 = 3 > \delta^2$$

Konstantin Busch - LSU

38

Tüme varım hipotezi: $3 \leq n \leq k$

Olduğunu varsayalım $f_n > \delta^{n-2}$

Tümevarım basamak: $n = k + 1$

$4 \leq k$ için $f_{k+1} > \delta^{(k-1)}$ olduğunu ispatlayalım

δ için $x^2 - x - 1 = 0$ denklemini çözelim



$$\delta^2 = \delta + 1$$



$$\delta^{k-1} = \delta^2 \cdot \delta^{k-3} = (\delta + 1)\delta^{k-3} = \delta^{k-2} + \delta^{k-3}$$



$$f_{k+1} = f_k + f_{k-1} \geq \delta^{k-2} + \delta^{k-3} = \delta^{k-1}$$

Tümevarım hipotezi

İspat sonu

En büyük ortak bölen

Özyinelemeli temel: $\text{gcd}(a, 0) = a$

Özyinelemeli basamak:

$$\text{gcd}(a, b) = \text{gcd}(b, a \bmod b)$$
$$a > b$$

En büyük ortak bölen için özyinelemeli algoritma

```
gcd(a, b) { //a>b olduğunu varsayalım
  if b = 0 then //özyineleme temel
    return a
  else //özyineleme basamak
    return gcd(b, a mod b)
}
```

Lamés Teoremi:

$\gcd(a, b)$, $a \geq b$ için Euclidian algoritması
en çok $5 \cdot \log_{10} b$ bölerek (iterasyon) kullanılır.

İspat:

Algoritmanın bölümünde Fibonacci
ilişkilerinde göstermiştik.

Konstantin Busch - LSU

43

Bölümler	$a = r_0$	$b = r_1$	kalanlar
r_0 / r_1	r_0	$= r_1 q_1 + r_2$	$0 < r_2 < r_1$
r_1 / r_2	r_1	$= r_2 q_2 + r_3$	$0 < r_3 < r_2$
$\vdots$	$\vdots$		$\vdots$
r_{n-2} / r_{n-1}	r_{n-2}	$= r_{n-1} q_{n-1} + r_n$	$0 < r_n < r_{n-1}$
r_{n-1} / r_n	r_{n-1}	$= r_n q_n + 0$	

İlk sıfır

sonuç

$$\gcd(a, b) = \gcd(r_0, r_1) = \gcd(r_1, r_2) = \gcd(r_2, r_3) \cdots$$
$$\cdots = \gcd(r_{n-2}, r_{n-1}) = \gcd(r_{n-1}, r_n) = \gcd(r_n, 0) = r_n$$

Konstantin Busch - LSU

44

$$\begin{aligned}
r_0 &= r_1 q_1 + r_2 \Rightarrow r_0 \geq r_1 + r_2 \\
r_1 &= r_2 q_2 + r_3 \Rightarrow r_1 \geq r_2 + r_3 \\
&\vdots \\
r_{n-2} &= r_{n-1} q_{n-1} + r_n \Rightarrow r_{n-2} \geq r_{n-1} + r_n \\
r_{n-1} &= r_n q_n + 0 \Rightarrow r_{n-1} \geq 2r_n
\end{aligned}$$

$r_n < r_{n-1}$ ve q_n tam sayı olduğu için burada sonlanır.

$r_n = \gcd(a, b) \geq 1$ olduğu için burada kalır.

$$r_n \geq 1$$

$$r_{n-1} \geq 2r_n$$

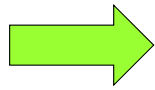
$$r_{n-2} \geq r_{n-1} + r_n$$

$$r_{n-3} \geq r_{n-2} + r_{n-1}$$

$\vdots$

$$r_2 \geq r_3 + r_4$$

$$r_1 \geq r_2 + r_3$$



$$r_n \geq 1 = f_2$$

$$r_{n-1} \geq 2r_n \geq 2f_2 = 2 = f_3$$

$$r_{n-2} \geq r_{n-1} + r_n \geq f_3 + f_2 = f_4$$

$$r_{n-3} \geq r_{n-2} + r_{n-1} \geq f_4 + f_3 = f_5$$

$\vdots$

$$r_2 \geq r_3 + r_4 \geq f_{n-1} + f_{n-2} = f_n$$

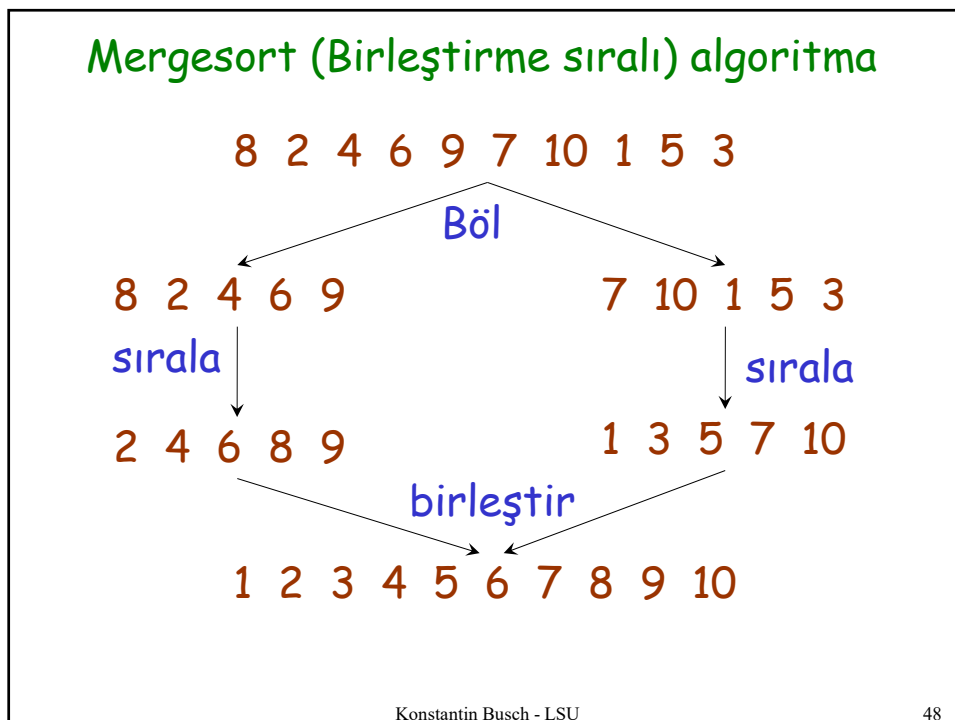
$$r_1 \geq r_2 + r_3 \geq f_n + f_{n-1} = f_{n+1}$$

$$\begin{array}{l}
 \left. \begin{array}{l} b = r_1 \geq f_{n+1} \\ f_{n+1} > \delta^{n-1} \end{array} \right\} \longrightarrow b > \delta^{n-1} \\
 \downarrow \\
 \log_{10} b > (n-1) \log_{10} \delta \\
 \downarrow \\
 n < \frac{\log_{10} b}{\log_{10} \delta} + 1 < 5 \cdot \log_{10} b + 1 \\
 \downarrow \\
 n \leq 5 \cdot \log_{10} b
 \end{array}$$

İspat sonu

Konstantin Busch - LSU

47




```

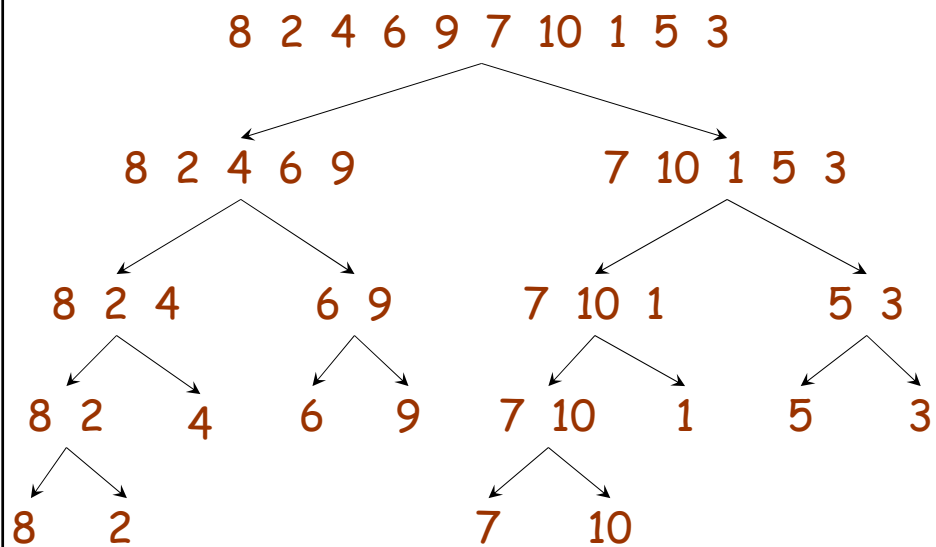
sort( $a_1, a_2, \dots, a_n$ ) {
  if  $n > 1$  then {
     $m = \lfloor n/2 \rfloor$ 
     $A \leftarrow \text{sort}(a_1, a_2, \dots, a_m)$ 
     $B \leftarrow \text{sort}(a_m, a_{m+1}, \dots, a_n)$ 
    return merge( $A, B$ )
  }
  else return  $a_1$ 
}

```

Konstantin Busch - LSU

49

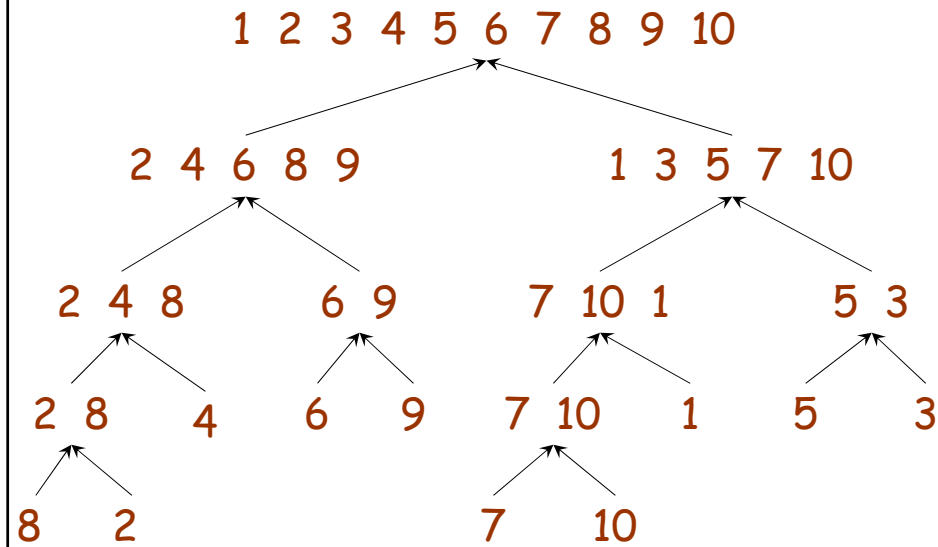
Özyinelemeli çağrılarının giriş değerleri



Konstantin Busch - LSU

50

Birleştirmenin giriş ve çıkış değerleri



Konstantin Busch - LSU

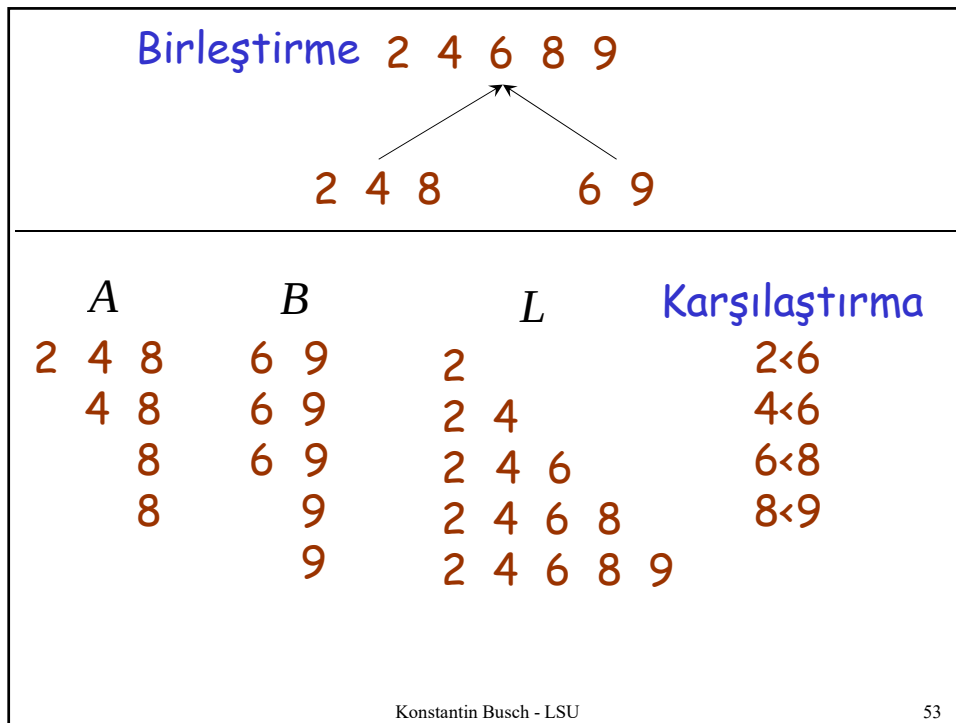
51

```

merge( A, B ) { //iki sıralanmış liste
    L ← ∅
    while A ≠ ∅ and B ≠ ∅ do {
        Listedeki A, B en küçük eleman
        çıkartılır ve L eklenir
    }
    if A ≠ ∅ or B ≠ ∅ then {
        L kalan elemanlar eklenir
    }
    return L
}
    
```

Konstantin Busch - LSU

52



A, B İki listeyi birleştirmek için karşılaştırılan toplam en çok sayı;

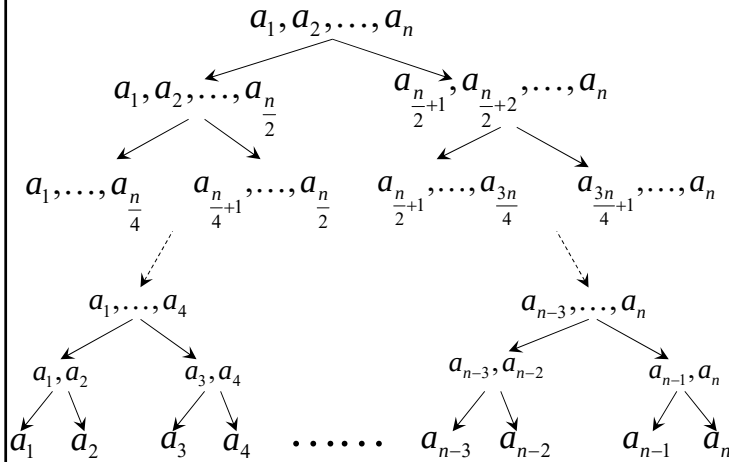
Birleştirme büyüklüğü

$$\# \text{ comparisons} \leq |A| + |B|$$

$\nearrow$
A'nın uzunluğu
 $\nwarrow$
B'nin uzunluğu

Konstantin Busch - LSU 54

Özyinelemeli çağırma ağacı

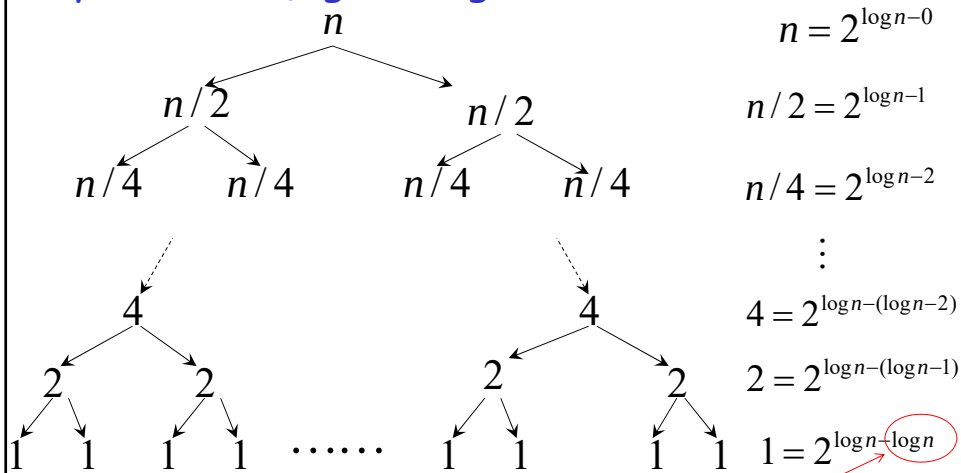


Varsayım olarak
 $n = 2^k$

Konstantin Busch - LSU

55

Özyinelemeli çağırma ağacı Liste başına elemanlar

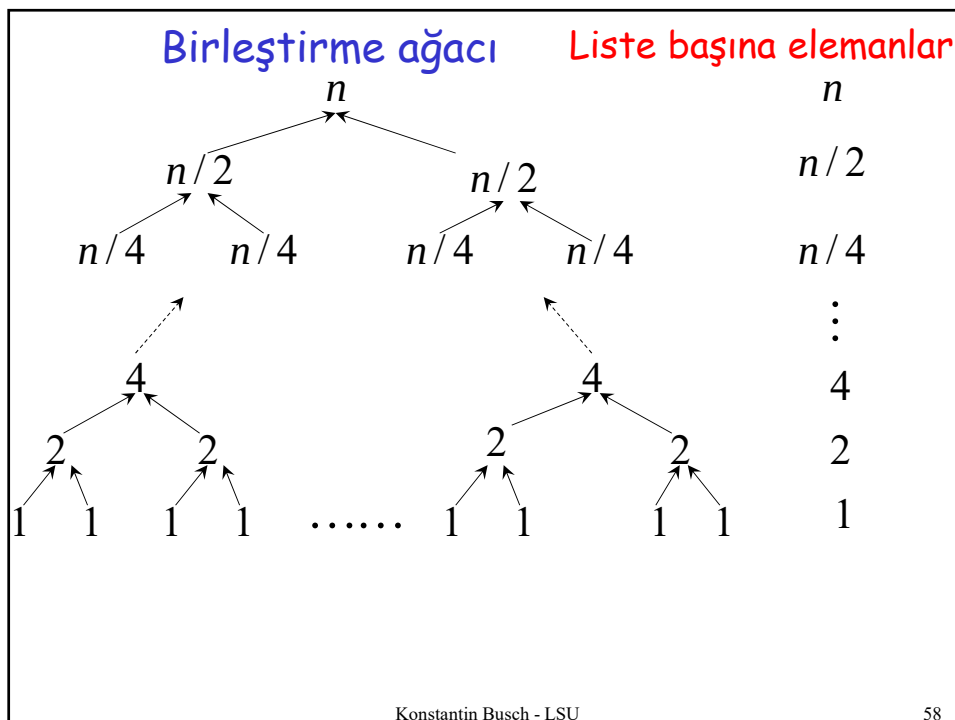
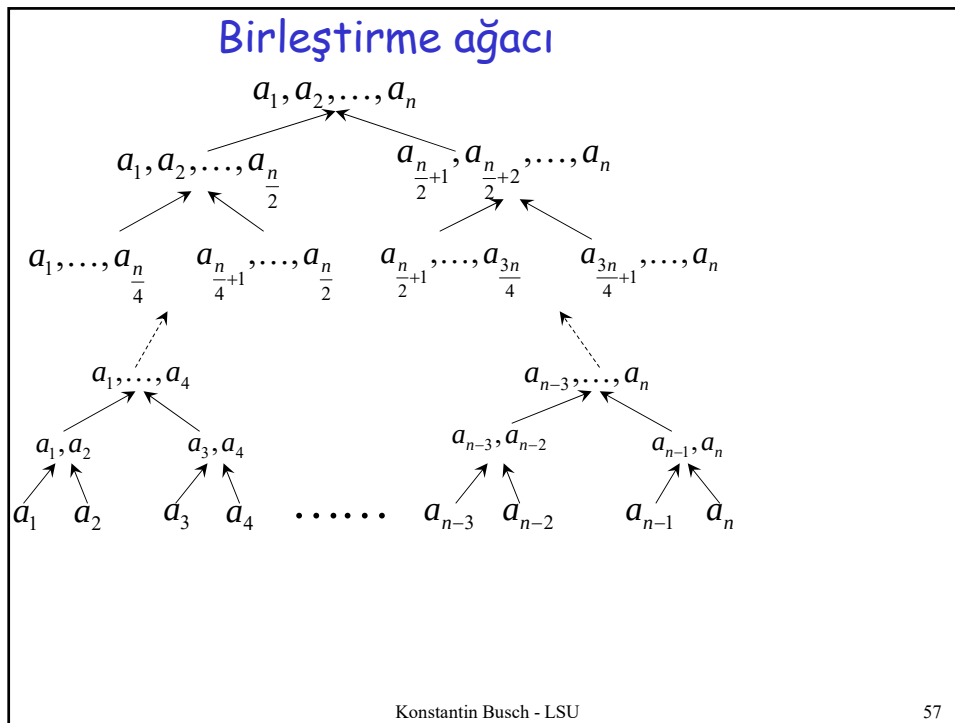


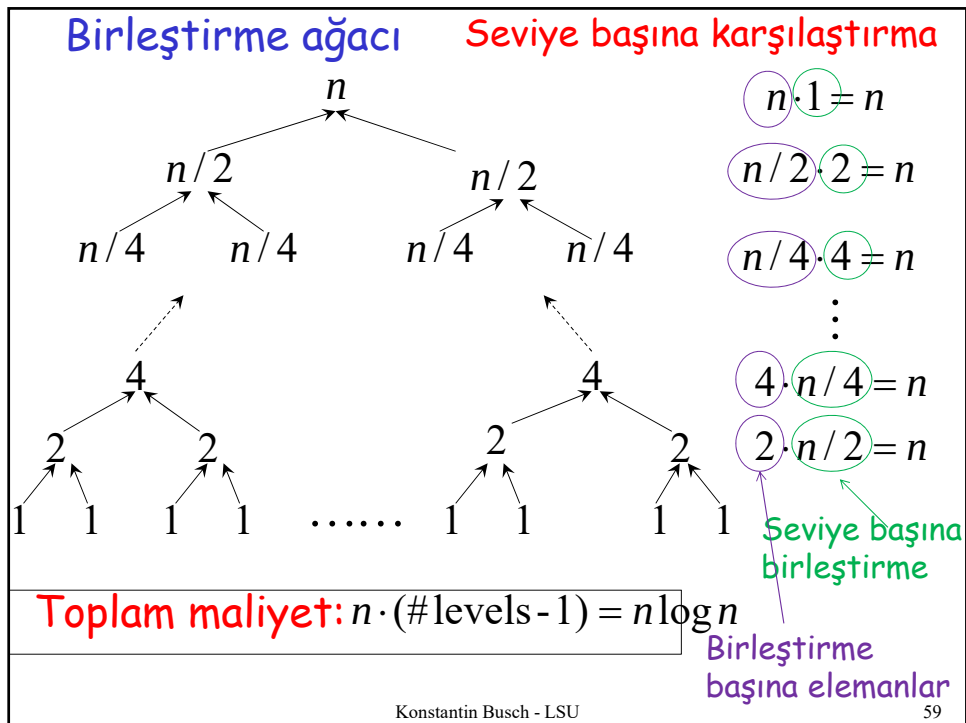
Varsayım
 $n = 2^k$

#ağacın seviyesi = $1 + \log n$

Konstantin Busch - LSU

56





Eğer karşılaştırma sayısı $n = 2^k$ ise
en çok $n \log n$

Eğer karşılaştırma sayısı $n \neq 2^k$ ise
en çok $m \log m = O(n \log n)$

$$m = 2^{\lceil \log n \rceil} < 2n$$

Birleştirmeli sıralamanın zaman karmaşıklığı:
 $O(n \log n)$

Konstantin Busch - LSU

60