Chapter 5 Divide and Conquer

BÖL ve YÖNET

2 Farklı Sıralama Algoritması

- Insertion Sort (Araya Eklemeli)
- Merge Sort (Birleştirme Sıralaması)

Insertion Sort (Araya Eklemeli)

	5	**2**	**1**	**3**	**7**
n:	1	2	3	4	5

Divide and Conquer and Merge

Böl Yönet Birleştir

Tanım

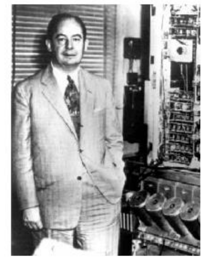
Çalışma zamanı analizi – Rekürans ilişkisi

Uygulama alanları

Zaman Karmaşıklığı

ORNEK 1) Mergesort.

- Divide array into two halves.
- Recursively sort each half.
- Merge two halves to make sorted whole.



Jon von Neumann (1945)

A	L	G	O	R	I	T	H	M	S

A	L	G	O	R	I	T	H	M	S	divide	$O(1)$
A	G	L	O	R	H	I	M	S	T	sort	$2T(n/2)$

A	G	H	I	L	M	O	R	S	T	merge	$O(n)$

T(n): n boyutlu bir girdide Mergesort için iki elemanı karşılaştırma sayısı

Mergesort için rekürans bağıntısı:

Mergesort recurrence.

$$T(n) \le \begin{cases} 0 & \text{if } n = 1 \\ \underbrace{T(\lceil n/2 \rceil)}_{\text{solve left half}} + \underbrace{T(\lfloor n/2 \rfloor)}_{\text{solve right half}} + \underbrace{n}_{\text{merging}} & \text{otherwise} \end{cases}$$

Solution. $T(n) = O(n \log_2 n)$.

Örnek:

6 5 3 1 8 7 2 4

Birlestirme Siralamasi (Merge Sort) ile cozum:

Mergesort recurrence.

$$T(n) \le \begin{cases} 0 & \text{if } n = 1 \\ \underbrace{T(\lceil n/2 \rceil)}_{\text{solve left half}} + \underbrace{T(\lfloor n/2 \rfloor)}_{\text{solve right half}} + \underbrace{n}_{\text{merging}} & \text{otherwise} \end{cases}$$

Solution. $T(n) = O(n \log_2 n)$.

İspatı

1) Özyineleme Ağacıyla(Grafiksel)

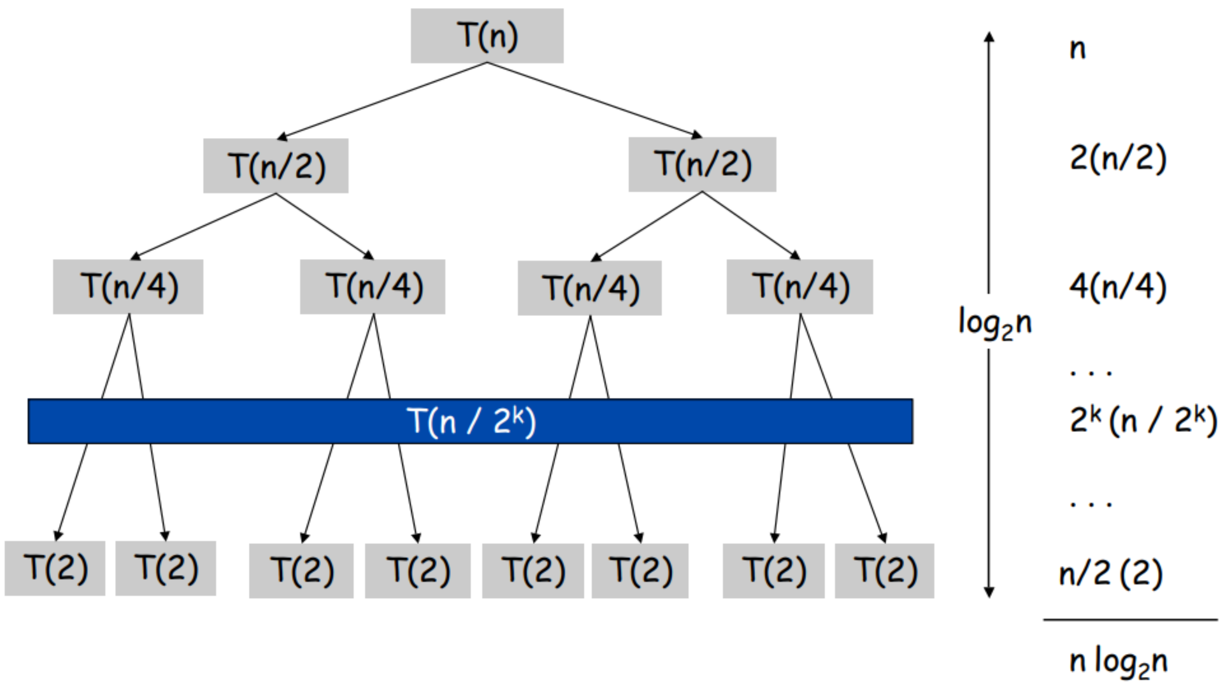
2) Teleskop yöntemi ile (seri çözümü)

3)Tümevarım ile

4) Master Teorem ile

1) Özyineleme Ağacıyla(Grafiksel)

$$T(n) = \begin{cases} 0 & \text{if } n = 1 \\ \underbrace{2T(n/2)}_{\text{sorting both halves}} + \underbrace{n}_{\text{merging}} & \text{otherwise} \end{cases}$$



2)Teleskop yöntemi ile (seri çözümü)

Teleskop yöntemi, matematikte bir serinin yakınsama veya dağılma durumunu kanıtlamak için kullanılan bir yöntemdir ve ardışık terimleri iptal ederek seriyi basitleştirmek ve yalnızca birkaç terimi bırakmaktır. "Teleskop" terimi, serinin bir çözülmeye veya "teleskobik" bir ifadeye indirgenebileceği benzetmesinden gelir.

Pf. For n > 1:

$$\frac{T(n)}{n} = \frac{2T(n/2)}{n} + 1$$

3)Tümevarım ile

$$T(n) = \begin{cases} 0 & \text{if } n = 1 \\ \underbrace{2T(n/2)}_{\text{sorting both halves}} + \underbrace{n}_{\text{merging}} & \text{otherwise} \end{cases}$$

Pf. (by induction on n)

- Base case: $n = 1$.
- Inductive hypothesis: $T(n) = n \log_2 n$.
- Goal: show that $T(2n) = 2n \log_2 (2n)$.

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

4) Master Teorem ile

Master Theorem for Recurrences of the Form
$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$ **where** $a \geq 1, b > 1$

$f(n)$	Conditions	$T(n)$
$O\left(n^{(\log_b a)-x}\right)$	$x > 0$	$O\left(n^{(\log_b a)}\right)$
$O\left(n^{(\log_b a)} \cdot (\log_2 n)^k\right)$	$k \geq 0$	$O\left(n^{(\log_b a)} \cdot (\log_2 n)^{k+1}\right)$
$O\left(n^{(\log_b a)+x}\right)$	$x > 0$ $a \cdot f\left(\frac{n}{b}\right) \leq c \cdot f(n)$ $c < 1$	$O(f(n))$

a) $T(n) = 4 \cdot T\left(\frac{n}{2}\right) + n$

b) $T(n) = 4 \cdot T\left(\frac{n}{2}\right) + n^2$

c) $T(n) = 4 \cdot T\left(\frac{n}{2}\right) + n^3$

ORNEK 2) Counting Inversions

Farklılıkların sayısı

İnversiyon sayısını hesaplamak, genellikle sıralama algoritmaları bağlamında karşımıza çıkan bir kavramdır ve verilen bir dizinin ne kadar düzensiz veya sırasız olduğunu ölçmek için kullanılır. Bir dizide inversiyon, iki öğenin yanlış sıralanmış olduğu durumu ifade eder.

Bu, bir dizideki elemanların sırasızlığını ölçen bir kavramdır, yani iki elemanın sırasının doğru olmaması durumu.

Örnek :

BAZ sıralama: A B C D E

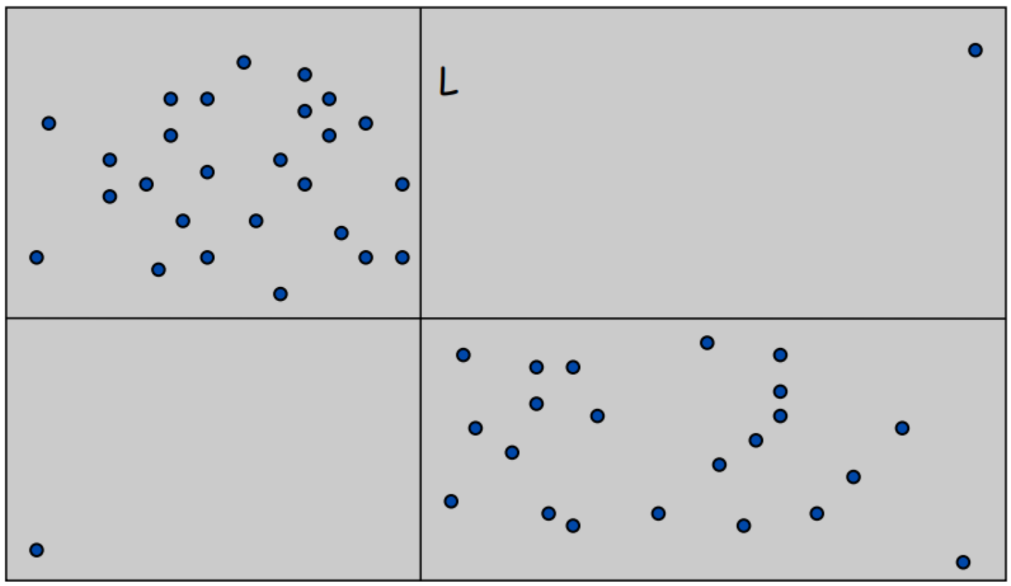
DİĞER : E A B D C

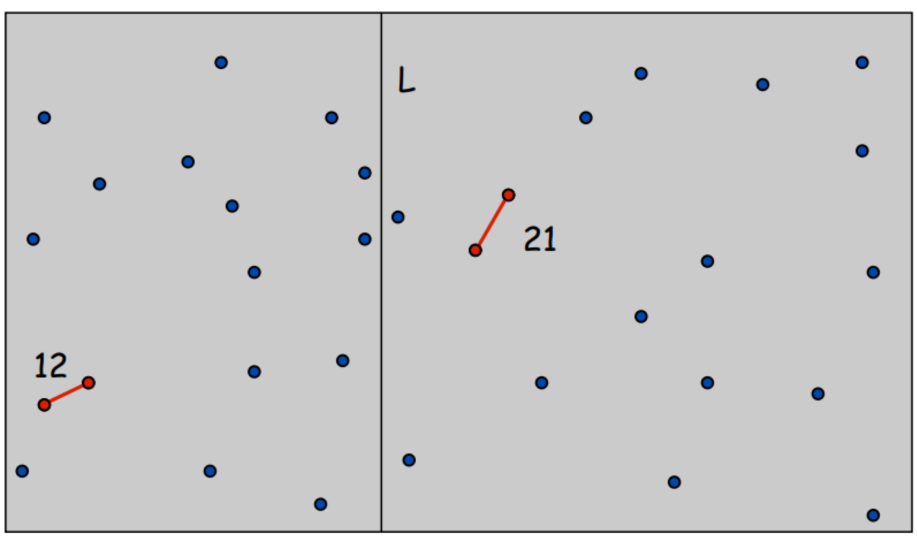
Örnek : Böl ve Yönet Algoritmasını kullanmadan ve kullanarak farka bakalım : O(?)

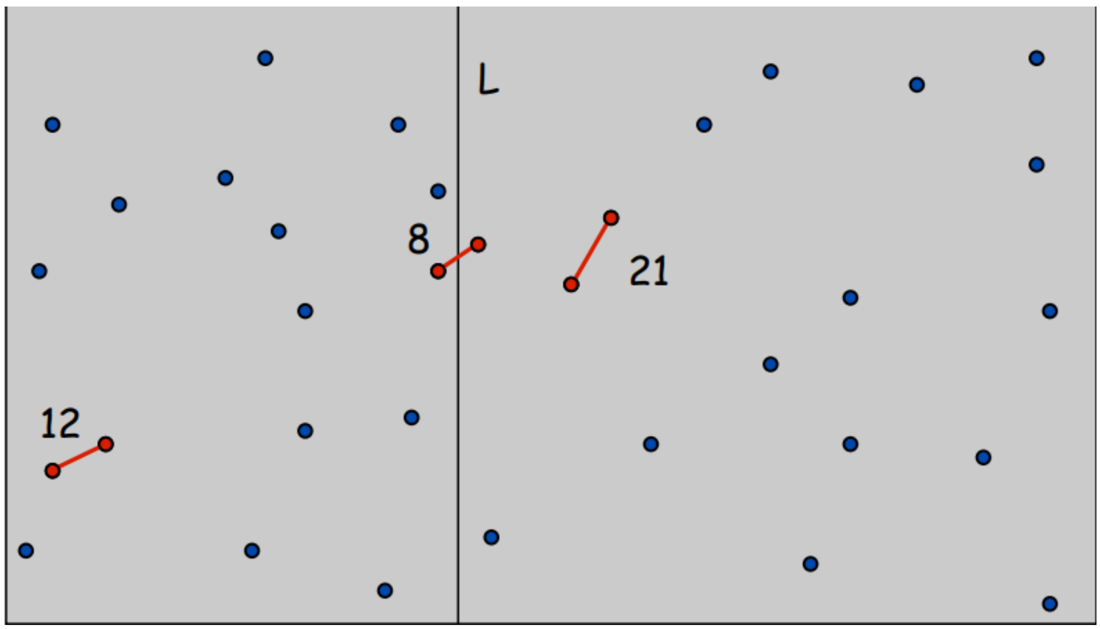
1 5 4 8 7 2 6 9

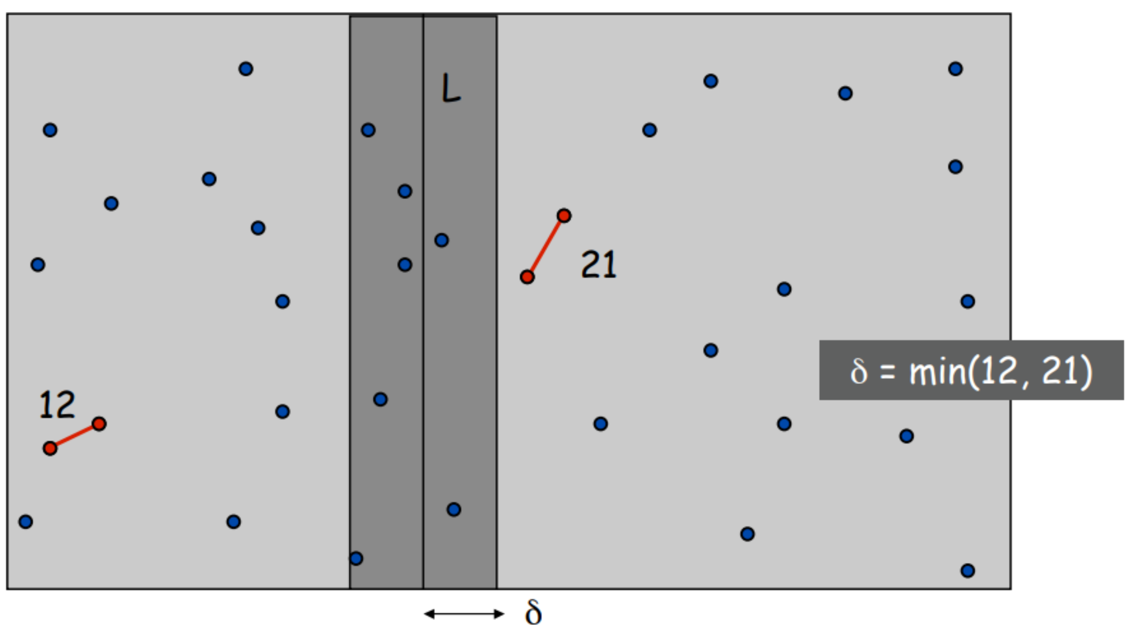
ORNEK 3) Closest Pair of Points

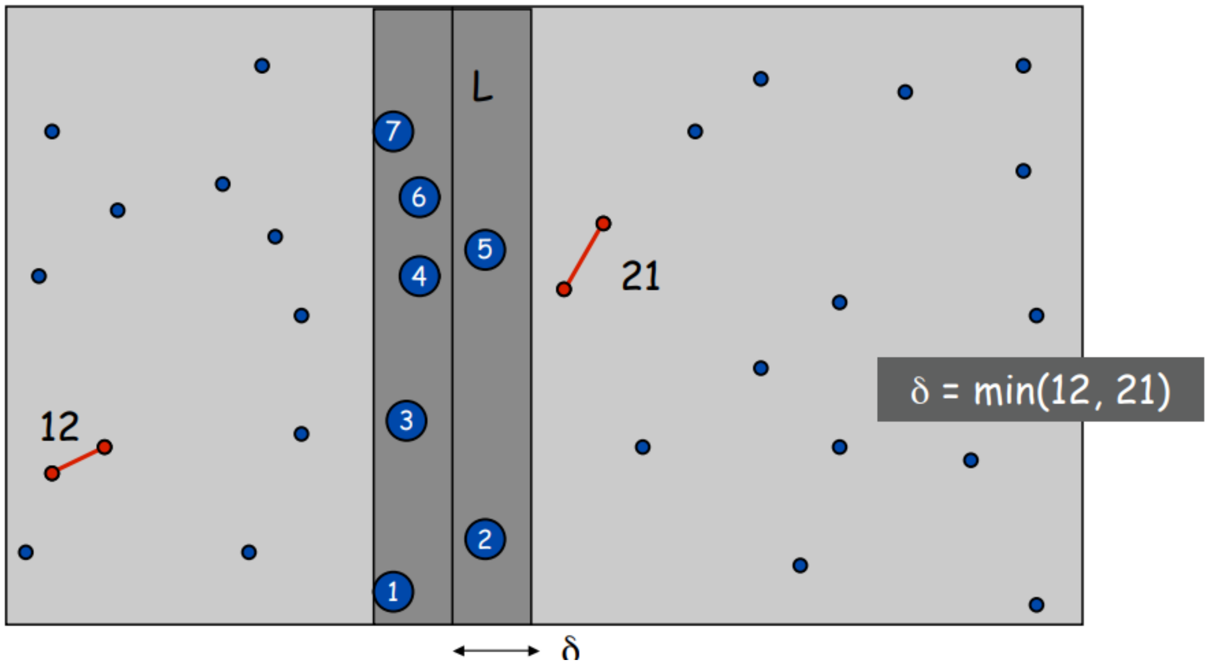
En Yakın Nokta Çifti






L
8
21
12


L
21
12
δ = min(12, 21)
δ


L
7
6
5
4
3
2
1
21
12
δ = min(12, 21)
δ