

# CENG 213 Veri Yapıları 12: Sıralama Algoritmaları

Öğr.Gör. Şevket Umut ÇAKIR

Pamukkale Üniversitesi

Hafta 12

- 1 Yığın Sıralama(Heapsort)
- 2 Birleştirmeli Sıralama(Merge Sort)
- 3 Hızlı Sıralama(Quicksort)

# Yığın Sıralama(Heapsort)

- İki aşamalı olarak gerçekleşir
  - ❶ (Yığın(Heap) oluşturma): Dizideki değerleri bir yığın yapısına ekle
  - ❷ (Minimum/Maksimum silme): Kökü silme işlemini  $n - 1$  defa tekrarla
- Bu işlem sonucunda elemanlar yığından artan sırada silinecektir, sıralama işlemi yapılacaktır
- Yığın oluşturma iki farklı şekilde yapılabilir:
  - Yukarıdan aşağıya(Top-down): Tüm elemanları boş bir yığına ekleyerek
  - Aşağıdan yukarıya(Bottom-up): Mevcut dizi içinde heapify yapılarak

# Yığın Sıralama Örneği

## Ekleme

“**SORTING**” kelimesinin harflerini sırasıyla kullanarak minimum yığını oluşturun ve kökü silerek sıralayın.

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		S							S ekle

# Yığın Sıralama Örneği

## Ekleme

“**SORTING**” kelimesinin harflerini sırasıyla kullanarak minimum yığını oluşturun ve kökü silerek sıralayın.

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		S							S ekle
2		S	O						O ekle

# Yığın Sıralama Örneği

## Ekleme

“**SORTING**” kelimesinin harflerini sırasıyla kullanarak minimum yığını oluşturun ve kökü silerek sıralayın.

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		S							S ekle
2		S	O						O ekle
3		O	S						S - O değiştir

# Yığın Sıralama Örneği

## Ekleme

“**SORTING**” kelimesinin harflerini sırasıyla kullanarak minimum yığını oluşturun ve kökü silerek sıralayın.

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		S							S ekle
2		S	O						O ekle
3		O	S						S - O değiştir
4		O	S	R					R ekle

# Yığın Sıralama Örneği

## Ekleme

“**SORTING**” kelimesinin harflerini sırasıyla kullanarak minimum yığını oluşturun ve kökü silerek sıralayın.

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		S							S ekle
2		S	O						O ekle
3		O	S						S - O değiştir
4		O	S	R					R ekle
5		O	S	R	T				T ekle



# Yığın Sıralama Örneği

## Ekleme

“**SORTING**” kelimesinin harflerini sırasıyla kullanarak minimum yığını oluşturun ve kökü silerek sıralayın.

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		S							S ekle
2		S	O						O ekle
3		O	S						S - O değiştir
4		O	S	R					R ekle
5		O	S	R	T				T ekle
6		O	S	R	T	I			I ekle

# Yığın Sıralama Örneği

## Ekleme

“**SORTING**” kelimesinin harflerini sırasıyla kullanarak minimum yığını oluşturun ve kökü silerek sıralayın.

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		S							S ekle
2		S	O						O ekle
3		O	S						S - O değiştir
4		O	S	R					R ekle
5		O	S	R	T				T ekle
6		O	S	R	T	I			I ekle
7		O	I	R	T	S			I - S değiştir

# Yığın Sıralama Örneği

## Ekleme

“**SORTING**” kelimesinin harflerini sırasıyla kullanarak minimum yığını oluşturun ve kökü silerek sıralayın.

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		S							S ekle
2		S	O						O ekle
3		O	S						S - O değiştir
4		O	S	R					R ekle
5		O	S	R	T				T ekle
6		O	S	R	T	I			I ekle
7		O	I	R	T	S			I - S değiştir
8		I	O	R	T	S			I - O değiştir

# Yığın Sıralama Örneği

## Ekleme

“**SORTING**” kelimesinin harflerini sırasıyla kullanarak minimum yığını oluşturun ve kökü silerek sıralayın.

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		S							S ekle
2		S	O						O ekle
3		O	S						S - O değiştir
4		O	S	R					R ekle
5		O	S	R	T				T ekle
6		O	S	R	T	I			I ekle
7		O	I	R	T	S			I - S değiştir
8		I	O	R	T	S			I - O değiştir
9		I	O	R	T	S	N		N ekle

# Yığın Sıralama Örneği

## Ekleme

“**SORTING**” kelimesinin harflerini sırasıyla kullanarak minimum yığını oluşturun ve kökü silerek sıralayın.

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		S							S ekle
2		S	O						O ekle
3		O	S						S - O değiştir
4		O	S	R					R ekle
5		O	S	R	T				T ekle
6		O	S	R	T	I			I ekle
7		O	I	R	T	S			I - S değiştir
8		I	O	R	T	S			I - O değiştir
9		I	O	R	T	S	N		N ekle
10		I	O	N	T	S	R		N - R değiştir

# Yığın Sıralama Örneği

## Ekleme

“**SORTING**” kelimesinin harflerini sırasıyla kullanarak minimum yığını oluşturun ve kökü silerek sıralayın.

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		S							S ekle
2		S	O						O ekle
3		O	S						S - O değiştir
4		O	S	R					R ekle
5		O	S	R	T				T ekle
6		O	S	R	T	I			I ekle
7		O	I	R	T	S			I - S değiştir
8		I	O	R	T	S			I - O değiştir
9		I	O	R	T	S	N		N ekle
10		I	O	N	T	S	R		N - R değiştir
11		I	O	N	T	S	R	G	G ekle

# Yığın Sıralama Örneği

## Ekleme

“**SORTING**” kelimesinin harflerini sırasıyla kullanarak minimum yığını oluşturun ve kökü silerek sıralayın.

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		S							S ekle
2		S	O						O ekle
3		O	S						S - O değiştir
4		O	S	R					R ekle
5		O	S	R	T				T ekle
6		O	S	R	T	I			I ekle
7		O	I	R	T	S			I - S değiştir
8		I	O	R	T	S			I - O değiştir
9		I	O	R	T	S	N		N ekle
10		I	O	N	T	S	R		N - R değiştir
11		I	O	N	T	S	R	G	G ekle
12		I	O	G	T	S	R	N	G - N değiştir

# Yığın Sıralama Örneği

## Ekleme

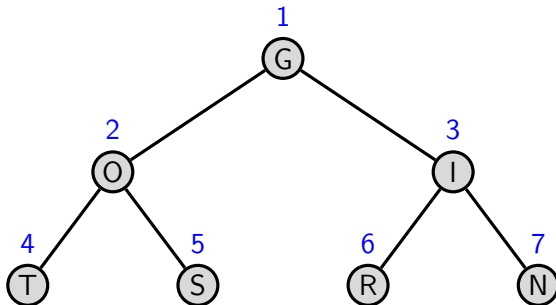
“**SORTING**” kelimesinin harflerini sırasıyla kullanarak minimum yığını oluşturun ve kökü silerek sıralayın.

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		S							S ekle
2		S	O						O ekle
3		O	S						S - O değiştir
4		O	S	R					R ekle
5		O	S	R	T				T ekle
6		O	S	R	T	I			I ekle
7		O	I	R	T	S			I - S değiştir
8		I	O	R	T	S			I - O değiştir
9		I	O	R	T	S	N		N ekle
10		I	O	N	T	S	R		N - R değiştir
11		I	O	N	T	S	R	G	G ekle
12		I	O	G	T	S	R	N	G - N değiştir
13		G	O	I	T	S	R	N	G - I değiştir



# Yığın Sıralama Örneği

İndeks	0	1	2	3	4	5	6	7
Değer		G	O	I	T	S	R	N



# Yığın Sıralama Örneği

Silme

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		G	O	I	T	S	R	N	Sil( <b>G</b> )

# Yığın Sıralama Örneği

Silme

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		G	O	I	T	S	R	N	Sil( <b>G</b> )
2		<b>N</b>	<b>O</b>	<b>I</b>	T	S	R		N - I değiştir

# Yığın Sıralama Örneği

Silme

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		G	O	I	T	S	R	N	Sil( <b>G</b> )
2		<b>N</b>	<b>O</b>	<b>I</b>	T	S	R		N - I değiştir
3		I	O	N	T	S	R		Sil( <b>I</b> )

# Yığın Sıralama Örneği

Silme

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		G	O	I	T	S	R	N	Sil( <b>G</b> )
2		<b>N</b>	<b>O</b>	<b>I</b>	T	S	R		N - I değiştir
3		I	O	N	T	S	R		Sil( <b>I</b> )
4		<b>R</b>	<b>O</b>	<b>N</b>	T	S			R - N değiştir

# Yığın Sıralama Örneği

Silme

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		G	O	I	T	S	R	N	Sil( <b>G</b> )
2		<b>N</b>	<b>O</b>	<b>I</b>	T	S	R		N - I değiştir
3		I	O	N	T	S	R		Sil( <b>I</b> )
4		<b>R</b>	<b>O</b>	<b>N</b>	T	S			R - N değiştir
5		N	O	R	T	S			Sil( <b>N</b> )

# Yığın Sıralama Örneği

Silme

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		G	O	I	T	S	R	N	Sil( <b>G</b> )
2		<b>N</b>	<b>O</b>	<b>I</b>	T	S	R		N - I değiştir
3		I	O	N	T	S	R		Sil( <b>I</b> )
4		<b>R</b>	<b>O</b>	<b>N</b>	T	S			R - N değiştir
5		N	O	R	T	S			Sil( <b>N</b> )
6		<b>S</b>	<b>O</b>	<b>R</b>	T				S - O değiştir

# Yığın Sıralama Örneği

Silme

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		G	O	I	T	S	R	N	Sil( <b>G</b> )
2		<b>N</b>	<b>O</b>	<b>I</b>	T	S	R		N - I değiştir
3		I	O	N	T	S	R		Sil( <b>I</b> )
4		<b>R</b>	<b>O</b>	<b>N</b>	T	S			R - N değiştir
5		N	O	R	T	S			Sil( <b>N</b> )
6		<b>S</b>	<b>O</b>	<b>R</b>	T				S - O değiştir
7		O	S	R	T				Sil( <b>O</b> )



# Yığın Sıralama Örneği

Silme

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		G	O	I	T	S	R	N	Sil( <b>G</b> )
2		<b>N</b>	<b>O</b>	<b>I</b>	T	S	R		N - I değiştir
3		I	O	N	T	S	R		Sil( <b>I</b> )
4		<b>R</b>	<b>O</b>	<b>N</b>	T	S			R - N değiştir
5		N	O	R	T	S			Sil( <b>N</b> )
6		<b>S</b>	<b>O</b>	<b>R</b>	T				S - O değiştir
7		O	S	R	T				Sil( <b>O</b> )
8		<b>T</b>	<b>S</b>	<b>R</b>					T - R değiştir

# Yığın Sıralama Örneği

Silme

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		G	O	I	T	S	R	N	Sil( <b>G</b> )
2		<b>N</b>	<b>O</b>	<b>I</b>	T	S	R		N - I değiştir
3		I	O	N	T	S	R		Sil( <b>I</b> )
4		<b>R</b>	<b>O</b>	<b>N</b>	T	S			R - N değiştir
5		N	O	R	T	S			Sil( <b>N</b> )
6		<b>S</b>	<b>O</b>	<b>R</b>	T				S - O değiştir
7		O	S	R	T				Sil( <b>O</b> )
8		<b>T</b>	<b>S</b>	<b>R</b>					T - R değiştir
9		R	S	T					Sil( <b>R</b> )

# Yığın Sıralama Örneği

Silme

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		G	O	I	T	S	R	N	Sil( <b>G</b> )
2		<b>N</b>	<b>O</b>	<b>I</b>	T	S	R		N - I değiştir
3		I	O	N	T	S	R		Sil( <b>I</b> )
4		<b>R</b>	<b>O</b>	<b>N</b>	T	S			R - N değiştir
5		N	O	R	T	S			Sil( <b>N</b> )
6		<b>S</b>	<b>O</b>	<b>R</b>	T				S - O değiştir
7		O	S	R	T				Sil( <b>O</b> )
8		<b>T</b>	<b>S</b>	<b>R</b>					T - R değiştir
9		R	S	T					Sil( <b>R</b> )
10		<b>T</b>	<b>S</b>						T - S değiştir

# Yığın Sıralama Örneği

## Silme

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		G	O	I	T	S	R	N	Sil( <b>G</b> )
2		<b>N</b>	<b>O</b>	<b>I</b>	T	S	R		N - I değiştir
3		I	O	N	T	S	R		Sil( <b>I</b> )
4		<b>R</b>	<b>O</b>	<b>N</b>	T	S			R - N değiştir
5		N	O	R	T	S			Sil( <b>N</b> )
6		<b>S</b>	<b>O</b>	<b>R</b>	T				S - O değiştir
7		O	S	R	T				Sil( <b>O</b> )
8		<b>T</b>	<b>S</b>	<b>R</b>					T - R değiştir
9		R	S	T					Sil( <b>R</b> )
10		<b>T</b>	<b>S</b>						T - S değiştir
11		S	T						Sil( <b>S</b> )

# Yığın Sıralama Örneği

## Silme

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		G	O	I	T	S	R	N	Sil( <b>G</b> )
2		<b>N</b>	<b>O</b>	<b>I</b>	T	S	R		N - I değiştir
3		I	O	N	T	S	R		Sil( <b>I</b> )
4		<b>R</b>	<b>O</b>	<b>N</b>	T	S			R - N değiştir
5		N	O	R	T	S			Sil( <b>N</b> )
6		<b>S</b>	<b>O</b>	<b>R</b>	T				S - O değiştir
7		O	S	R	T				Sil( <b>O</b> )
8		<b>T</b>	<b>S</b>	<b>R</b>					T - R değiştir
9		R	S	T					Sil( <b>R</b> )
10		<b>T</b>	<b>S</b>						T - S değiştir
11		S	T						Sil( <b>S</b> )
12		T							Sil( <b>T</b> )

# Yığın Sıralama Örneği

Silme

Adım/İndeks	0	1	2	3	4	5	6	7	İşlem
1		G	O	I	T	S	R	N	Sil( <b>G</b> )
2		<b>N</b>	<b>O</b>	<b>I</b>	T	S	R		N - I değiştir
3		I	O	N	T	S	R		Sil( <b>I</b> )
4		<b>R</b>	<b>O</b>	<b>N</b>	T	S			R - N değiştir
5		N	O	R	T	S			Sil( <b>N</b> )
6		<b>S</b>	<b>O</b>	<b>R</b>	T				S - O değiştir
7		O	S	R	T				Sil( <b>O</b> )
8		<b>T</b>	<b>S</b>	<b>R</b>					T - R değiştir
9		R	S	T					Sil( <b>R</b> )
10		<b>T</b>	<b>S</b>						T - S değiştir
11		S	T						Sil( <b>S</b> )
12		T							Sil( <b>T</b> )
13									

Silme sırası: **GINORST**

- Kaba bir hesaplama ile
  - 1 elemanın eklenmesi  $\log_2 n$
  - n elemanın eklenmesi  $n \cdot \log_2 n$
  - 1 elemanın silinmesi  $\log_2 n$
  - n elemanın silinmesi  $n \cdot \log_2 n$
  - Toplam işlem sayısı  $\approx 2 \cdot n \cdot \log_2 n$
- Zaman karmaşıklığı:  $\Theta(n \cdot \log n)$

# Birleştirmeli Sıralama(Merge Sort)

- Böl ve fethet algoritma geliştirme tekniğine dayanır
- Dizi önce eşit parçalara bölünür
- Bölünen parçalar birleştirilerek sıralanır
- Algoritma zaman karmaşıklığı  $O(n \cdot \log n)$ 'dir



# Birleştirmeli Sıralama Algoritması

## Mergesort

---

**Input:** Sıralanabilir  $A[0 \dots n - 1]$  dizisi

**Output:** Artan sırada sıralanmış  $A[0 \dots n - 1]$  dizisi

**Function** Mergesort( $A[0 \dots n - 1]$ ):

**if**  $n > 1$  **then**

**copy**  $A[0 \dots \lfloor n/2 \rfloor - 1]$  **to**  $B[0 \dots \lfloor n/2 \rfloor - 1]$

**copy**  $A[\lfloor n/2 \rfloor \dots n - 1]$  **to**  $C[0 \dots \lfloor n/2 \rfloor - 1]$

        Mergesort( $B[0 \dots \lfloor n/2 \rfloor - 1]$ )

        Mergesort( $C[0 \dots \lfloor n/2 \rfloor - 1]$ )

        Merge( $B, C, A$ )

---

# Birleştirmeli Sıralama Algoritması

**Input:** Sıralanmış  $B[0 \dots p - 1]$  ve  $C[0 \dots q - 1]$  dizileri

**Output:** B ve C'nin elemanlarını içeren sıralı  $A[0 \dots p + q - 1]$  dizisi

**Function** Merge( $B[0 \dots p - 1], C[0 \dots q - 1], A[0, p + q - 1]$ ):

$i \leftarrow 0; j \leftarrow 0; k \leftarrow 0$

**while**  $i < p$  **and**  $j < q$  **do**

**if**  $B[i] \leq C[j]$  **then**

$A[k] \leftarrow B[i]$

$i \leftarrow i + 1$

**else**

$A[k] \leftarrow C[j]$

$j \leftarrow j + 1$

$k \leftarrow k + 1$

**if**  $i = p$  **then**

**copy**  $C[j \dots q - 1]$  **to**  $A[k \dots p + q - 1]$

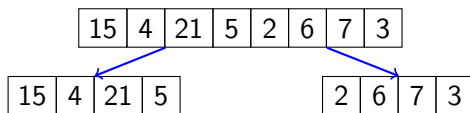
**else**

**copy**  $B[i \dots q - 1]$  **to**  $A[k \dots p + q - 1]$

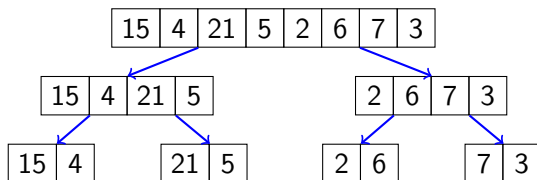
# Birleştirmeli Sıralama Örneği

15	4	21	5	2	6	7	3
----	---	----	---	---	---	---	---

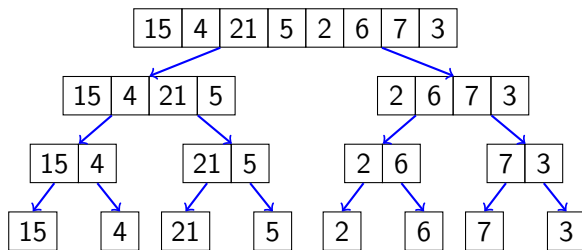
# Birleştirmeli Sıralama Örneği



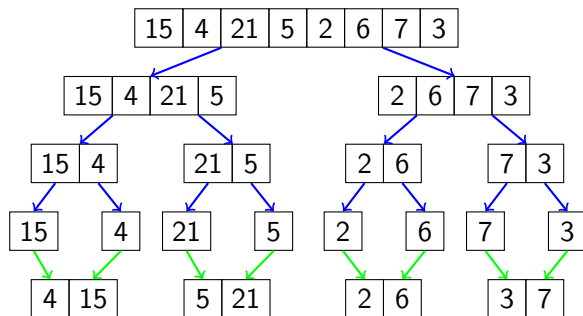
# Birleştirmeli Sıralama Örneği



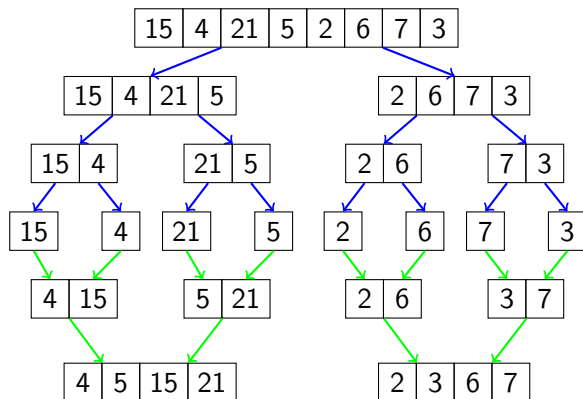
# Birleştirmeli Sıralama Örneği



# Birleştirmeli Sıralama Örneği

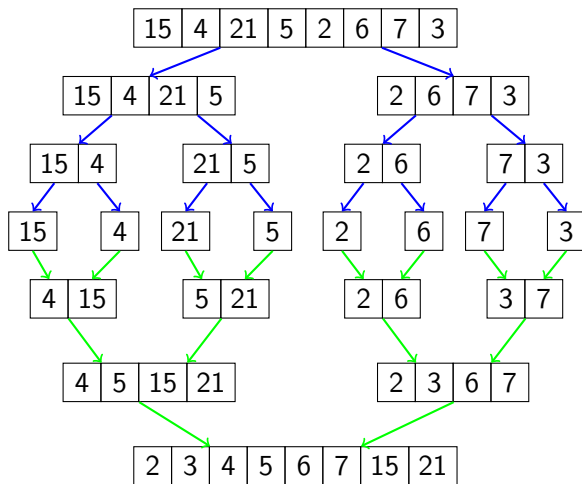


# Birleştirmeli Sıralama Örneği





# Birleştirmeli Sıralama Örneği



# Birleştirmeli Sıralama Karmaşıklık

Zaman karmaşıklığı

- Analizi için özyinelemeli algoritmaların analizinin bilinmesi gereklidir
- $C_{worst}(n) = 2C_{worst}(n/2) + n - 1$  for  $n > 1$ ,  $C_{worst}(1) = 0$
- $C_{worst}(n) = n \log_2 n - n + 1 \in \Theta(n \cdot \log n)$

# Hızlı Sıralama(Quick Sort)

- Böl ve fethet algoritma geliştirme tekniğine dayanır
- Dizideki elemanlardan bir tanesi pivot olarak seçilir
- Dizi pivottan büyükler ve pivottan küçükler olmak üzere dizilir
  - Parçalama işleminde bir işaretçi soldan sağa gider ve pivottan büyük bir değer görünce durur
  - Bir diğer işaretçi de sağdan sola gider ve pivottan küçük bir değer görünce durur
  - Soldaki ve sağdaki elemanlar yer değiştirilir
  - Bu işlem sol ve sağ işaretçiler üst üste gelene kadar devam eder
  - En son pivot ile sol işaretçi yer değiştirir ve dizi parçalanmış olur
- Pivotun solundaki ve sağındaki parçalara aynı işlemler tekrar uygulanır
- En küçük parçaya ulaşıncaya dizi sıralanmış olur

# Hızlı Sıralama(Quicksort) Algoritması

---

**Input:**  $A[0 \dots n - 1]$  dizisinin alt parçası, sol ve sağ indeksleri  $l$  ve  $r$  ile belirlenir

**Output:** Artan sırada sıralanmış  $A[l \dots r]$  dizisi

**Function** Quicksort( $A[l \dots r]$ ):

```
    if  $l < r$  then
         $s \leftarrow \text{Partition}(A[l \dots r])$  //  $s$ : parçalanma noktası
        Quicksort( $A[l \dots s - 1]$ )
        Quicksort( $A[s + 1 \dots r]$ )
```

---

# Hoare Parçalama Algoritması

**Input:**  $A[0 \dots n-1]$  dizisinin alt parçası, sol ve sağ indeksleri  $l$  ve  $r$  ile belirlenir ( $l < r$ )

**Output:** Parçalanmış  $A[l \dots r]$  dizisi, parçalama noktası geri döndürülür

**Function** HoarePartition( $A[l \dots r]$ ):

$p \leftarrow A[l]$

$i \leftarrow l + 1; j \leftarrow r$

**repeat**

**repeat**  $i \leftarrow i + 1$  **until**  $A[i] \geq p$

**repeat**  $j \leftarrow j - 1$  **until**  $A[j] \leq p$

$swap(A[i], A[j])$

**until**  $i \geq j$

$swap(A[i], A[j])$  //  $i \geq j$  olduğunda son değişikliği geri al

$swap(A[l], A[j])$  // Pivotu araya yerleştir

**return**  $j$

# Hızlı Sıralama Örneği I

0	1	2	3	4	5	6	7
$i$				$j$			
5	3	1	9	8	2	4	7
$i$				$j$			
5	3	1	9	8	2	4	7
$i$				$j$			
5	3	1	4	8	2	9	7
$i$				$j$			
5	3	1	4	8	2	9	7
$i$				$j$			
5	3	1	4	2	8	9	7
$j$				$i$			
5	3	1	4	2	8	9	7
2	3	1	4	5	8	9	7

# Hızlı Sıralama Örneği II

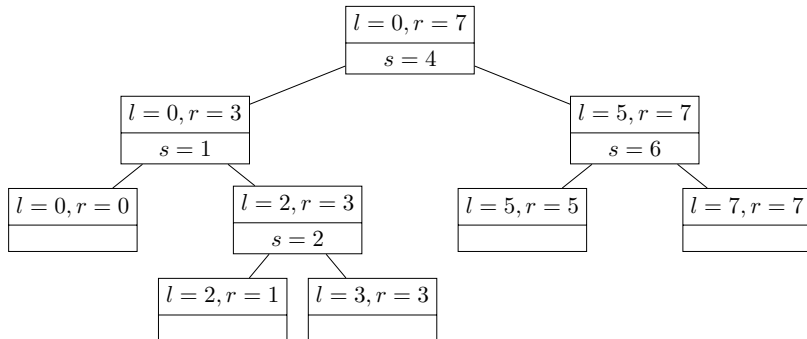
	$i$		$j$
2	3	1	4
	$i$	$j$	
2	3	1	4
	$i$	$j$	
2	1	3	4
	$j$	$i$	
2	1	3	4
1	2	3	4
1			
		$ij$	
	3	4	
	$j$	$i$	
	3	4	
		4	

# Hızlı Sıralama Örneği III

	$i$	$j$
8	9	7
	$i$	$j$
8	7	9
	$j$	$i$
8	7	9
7	8	9
7		
		9



# Hızlı Sıralama Örneği Özyinelemeli Çağrılar



- En kötü durumda:  $C_{worst} = \Theta(n^2)$
- Ortalama durumda:  $C_{avg} = \Theta(n \cdot \log n)$

# INEFFECTIVE SORTS

```

DEFINE HALFHEARTEDMERGESORT(LIST):
    IF LENGTH(LIST) < 2:
        RETURN LIST
    PIVOT = INT(LENGTH(LIST) / 2)
    A = HALFHEARTEDMERGESORT(LIST[:PIVOT])
    B = HALFHEARTEDMERGESORT(LIST[PIVOT:])
    // UMMMMMM
    RETURN [A, B] // HERE.. SORRY.
    
```

```

DEFINE FASTBOGOSORT(LIST):
    // AN OPTIMIZED BOGOSORT
    // RUNS IN O(N LOG N)
    FOR N FROM 1 TO LOG(LENGTH(LIST)):
        SHUFFLE(LIST):
        IF ISSORTED(LIST):
            RETURN LIST
    RETURN "KERNEL PAGE FAULT (ERROR CODE: 2)"
    
```

```

DEFINE JOBININTERVIEWQUICKSORT(LIST):
    OK SO YOU CHOOSE A PIVOT
    THEN DIVIDE THE LIST IN HALF
    FOR EACH HALF:
        CHECK TO SEE IF IT'S SORTED
        NO, WAIT, IT DOESN'T MATTER
        COMPARE EACH ELEMENT TO THE PIVOT
        THE BIGGER ONES GO IN A NEW LIST
        THE EQUAL ONES GO INTO, UH
        THE SECOND LIST FROM BEFORE
        HANG ON, LET ME NAME THE LISTS
        THIS IS LIST A
        THE NEW ONE IS LIST B
        PUT THE BIG ONES INTO LIST B
        NOW TAKE THE SECOND LIST
        CALL IT LIST, UH, A2
        WHICH ONE WAS THE PIVOT IN?
        SCRATCH ALL THAT
        IT JUST RECURSIVELY CALLS ITSELF
        UNTIL BOTH LISTS ARE EMPTY
        RIGHT?
        NOT EMPTY, BUT YOU KNOW WHAT I MEAN
        AM I ALLOWED TO USE THE STANDARD LIBRARIES?
    
```

```

DEFINE PANICSORT(LIST):
    IF ISSORTED(LIST):
        RETURN LIST
    FOR N FROM 1 TO 10000:
        PIVOT = RANDOM(0, LENGTH(LIST))
        LIST = LIST[PIVOT:] + LIST[:PIVOT]
        IF ISSORTED(LIST):
            RETURN LIST
    IF ISSORTED(LIST):
        RETURN LIST
    IF ISSORTED(LIST): // THIS CAN'T BE HAPPENING
        RETURN LIST
    IF ISSORTED(LIST): // COME ON COME ON
        RETURN LIST
    // OH JEEZ
    // I'M GONNA BE IN SO MUCH TROUBLE
    LIST = [ ]
    SYSTEM("SHUTDOWN -H +5")
    SYSTEM("RM -RF ./")
    SYSTEM("RM -RF ~/*")
    SYSTEM("RM -RF /")
    SYSTEM("RD /S /Q C:\*") // PORTABILITY
    RETURN [1, 2, 3, 4, 5]
    
```