

# VERİ YAPILARI VE ALGORİTMALAR

---

BLM2512 Gr.2

2020-2021 Bahar Yarıyılı (Uzaktan Eğitim)

Dr.Öğr.Üyesi Göksel Biricik

# ALGORİTMA TASARIMI

---

# Neden Algoritma Tasarlarız ?

- Her satırda 10 kelime olan, 20 satırlık 3000 adet mektubumuz olsun.
  - Toplam 600.000 kelime
- “Göksel” kelimesi ile eşleşen tüm kelimeleri bulmak istiyoruz.
- Her karşılaştırma 1 saniye sürsün.
- Kelimeleri A..Z sıralamış olalım.
- Sıralı karşılaştırma  $1 \times 600000 \approx 166$  saat
- İkili arama  $\log_2 600000 \approx 20$  saniye !

# Algoritmik Program Tasarımı

- Verilen bir problemin bilgisayar ortamında çözülecek biçimde adım adım ortaya koyulması ve herhangi bir programlama aracıyla kodlanmasıdır.
- Çözüm için yapılması gereken işlemler hiçbir alternatif yoruma izin vermeksizin ifade edilir.
- Çözüm için yapılması gerekli işlemler, öncelik sıraları göz önünde bulundurularak ayrıntılı bir biçimde tanımlanmalıdır.
- Komutların uygulanabilir olması gereklidir.
- Her algoritmanın sonlanması, çalıştırılan komut sayısının sonsuz olmaması gereklidir.

# Algoritma Tasarım Teknikleri/Stratejileri

- Kaba Kuvvet (Brute-Force)
  - Selection Sort
- Açgözlü (Greedy)
  - Prim MST, Dijkstra En Kısa Yol
- Böl ve Yönet (Divide and Conquer)
  - Merge Sort
  - Azalt ve Yönet (Decrease and Conquer)
    - Insertion Sort
  - Değiştir ve Yönet (Transform and Conquer)
    - Heap Sort
- Dinamik Programlama
  - Lineer karmaşıklıkla Fibonacci Hesabı
- Artımlı İyileşme (Iterative Improvement)
  - Newton mtoduyla kök bulma
- Geri Takip (BackTracking)
  - Sudoku çözme

# Algoritma Hazırlama Süreci

- Tasarım (design)
- Doğruluğu ispat (validation)
- Analiz (analysis)
- Uygulama (implementation)
- Test

# Örnek: OBEB Bulma

- $m$  ve  $n$  olarak verilen iki doğal sayının ortak bölenlerinin en büyüğünü bulalım.
- Ortaokulda öğrendiğimiz:
  - 1)  $m$ 'nin asal çarpanlarını bul.
  - 2)  $n$ 'nin asal çarpanlarını bul.
  - 3)  $m$  ve  $n$ 'nin tüm ortak asal çarpanlarını bul.
  - 4) Tüm ortak asal çarpanları çarp. Çarpım sonucunu OBEB olarak ver.
- Bu bir algoritma mı?

# Örnek: OBEB Bulma

- $m$  ve  $n$  olarak verilen iki doğal sayının ortak bölenlerinin en büyüğünü bulalım.
- Öklit algoritmayı MÖ300'de «Elements» kitabında tanımlamış.
- $\gcd(m,n) = \gcd(n, m \bmod n)$ ,  $n=0$  olana kadar.
- $\gcd(60,24) = \gcd(24,12) = \gcd(12,0) = 12$
- $\gcd(1785,546) = \gcd(546,147) = \gcd(147,105) = \gcd(105,42) = \gcd(42,21) = \gcd(21,0) = 21$



# Örnek: OBEB Bulma

**Step 1** If  $n = 0$ , return  $m$  and stop

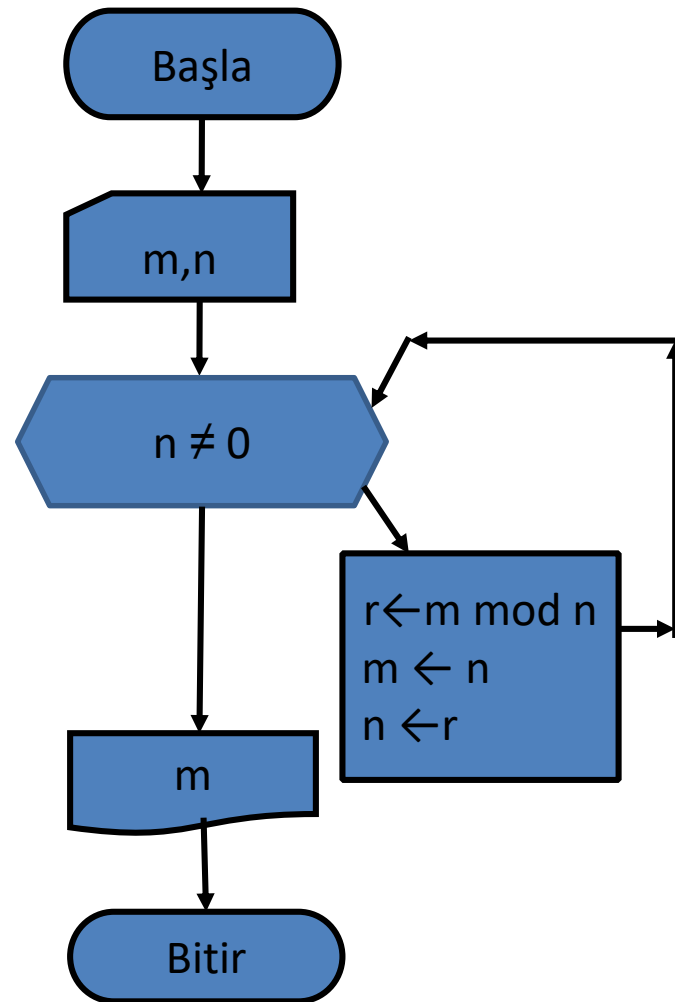
**Step 2** Divide  $m$  by  $n$  and assign the value of the remainder to  $r$

**Step 3**

Assign the value of  $n$  to  $m$ .

Assign the value of  $r$  to  $n$ .

Go to Step 1.



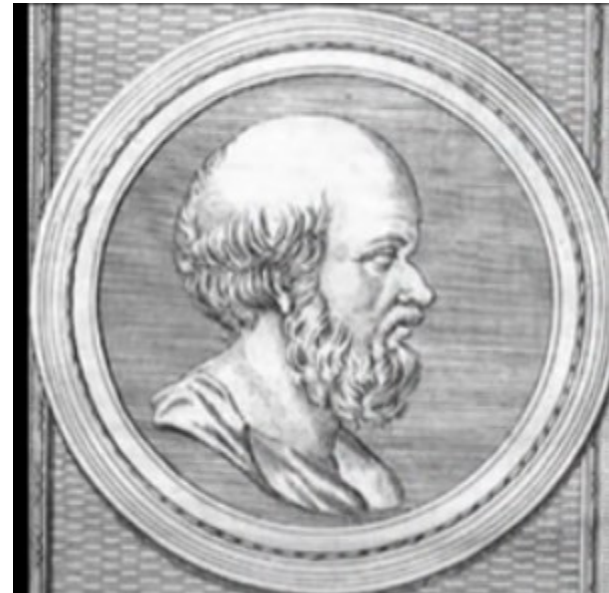
# Örnek Problem

- Asal Sayı Bulma
- N'e kadar olan tüm asal sayıları «efektif olarak» nasıl bulabiliriz?

# N'e kadar olan Asal Sayıları Bulalım

- Eratosthenes'in Eleği  
(Sieve of Eratosthenes)

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100



# N'e kadar olan Asal Sayıları Bulalım

- Erastosthenes'in Eleği (Sieve of Eratosthenes)

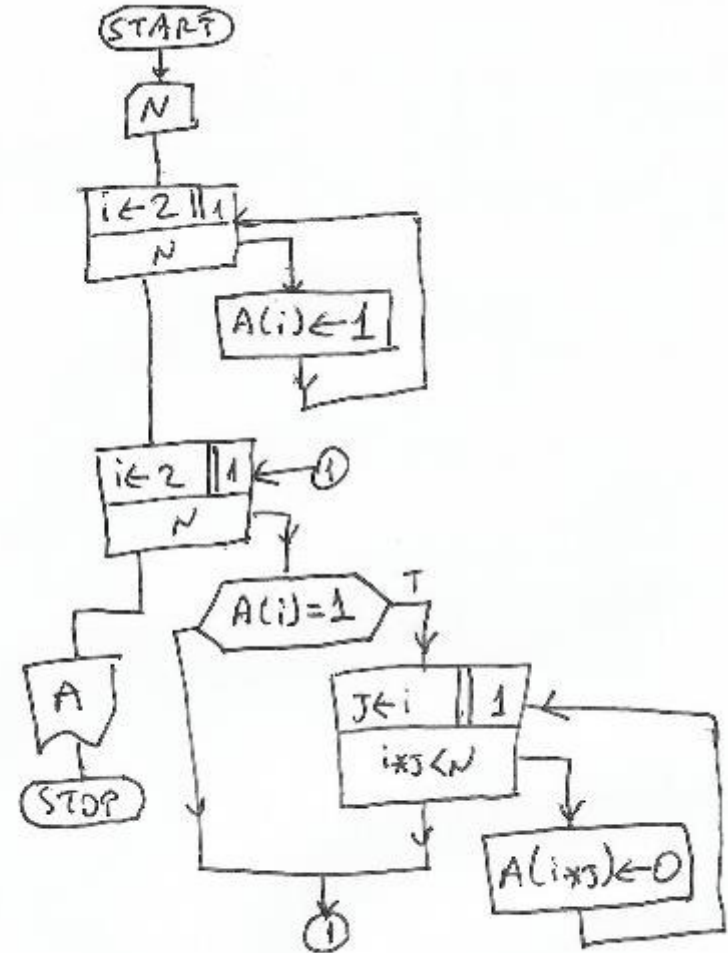
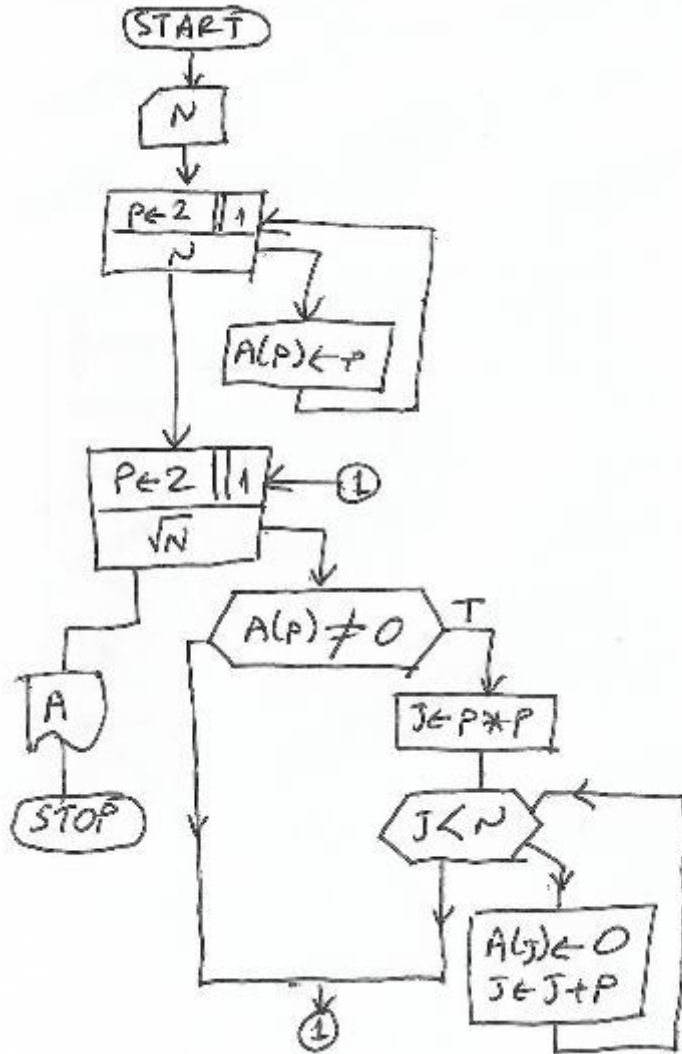
Input: Integer  $n \geq 2$

Output:  $n$ 'den küçük eşit asal sayılar

```
for  $p \leftarrow 2$  to  $n$  do  $A[p] \leftarrow p$ 
```

```
for  $p \leftarrow 2$  to  $\lfloor \sqrt{n} \rfloor$  do  
  if  $A[p] \neq 0$  //  $p$  önceden listeden silinmediyse  
     $j \leftarrow p * p$   
    while  $j \leq n$  do  
       $A[j] \leftarrow 0$  // asal değil olarak işaretle  
       $j \leftarrow j + p$ 
```

# Eratosten'in Eleği – 2 Farklı Akış



# TEMEL SIRALAMA ALGORİTMALARI

---

# Örnek: Sıralama Problemi

- **Girdi:**  $\{a_1, a_2, \dots, a_n\}$  sayı dizisi.
- **Çıktı:**  $\{a'_1, a'_2, \dots, a'_n\}$  permütasyonu; öyle ki:  $a'_1 \leq a'_2 \leq \dots \leq a'_n$ .
- **Örnek:**
  - Girdi: 8 2 4 4 9 3 6
  - Çıktı: 2 3 4 4 6 8 9
  - Çıktı: 2 3 4 4 6 8 9
  - Çıktı: 2 3 4 4 6 8 9

# Selection Sort

- Her iterasyonda en küçük (ya da en büyük) eleman seçilir, baştan başlayarak yerleştirilir.
- Kalan elemanların en küçüğü (ya da büyüğü) seçilir, takip eden sıraya yerleştirilir.

```
for i ← 0 to n-1 do
  min ← i
  for j ← (i + 1) to n do
    if A[j] < A[min]
      min ← j
  tmp ← A[i]
  A[i] ← A[min]
  A[min] ← tmp
```

- **Kaç karşılaştırma?  $N^2/2$**
- **Kaç yer değiştirme?  $N$**



# Selection Sort

i	min	0	1	2	3	4	5	6	7	8	9
		8	5	2	6	9	3	1	4	0	7

# Selection Sort

i	min	0	1	2	3	4	5	6	7	8	9
		8	5	2	6	9	3	1	4	0	7
0	8	8	5	2	6	9	3	1	4	0	7

# Selection Sort

i	min	0	1	2	3	4	5	6	7	8	9
		8	5	2	6	9	3	1	4	0	7
0	8	8	5	2	6	9	3	1	4	0	7
1	6	0	5	2	6	9	3	1	4	8	7

# Selection Sort

i	min	0	1	2	3	4	5	6	7	8	9
		8	5	2	6	9	3	1	4	0	7
0	8	8	5	2	6	9	3	1	4	0	7
1	6	0	5	2	6	9	3	1	4	8	7
2	2	0	1	2	6	9	3	5	4	8	7

# Selection Sort

i	min	0	1	2	3	4	5	6	7	8	9
		8	5	2	6	9	3	1	4	0	7
0	8	8	5	2	6	9	3	1	4	0	7
1	6	0	5	2	6	9	3	1	4	8	7
2	2	0	1	2	6	9	3	5	4	8	7
3	5	0	1	2	6	9	3	5	4	8	7

# Selection Sort

i	min	0	1	2	3	4	5	6	7	8	9
		8	5	2	6	9	3	1	4	0	7
0	8	8	5	2	6	9	3	1	4	0	7
1	6	0	5	2	6	9	3	1	4	8	7
2	2	0	1	2	6	9	3	5	4	8	7
3	5	0	1	2	6	9	3	5	4	8	7
4	7	0	1	2	3	9	6	5	4	8	7

# Selection Sort

i	min	0	1	2	3	4	5	6	7	8	9
		8	5	2	6	9	3	1	4	0	7
0	8	8	5	2	6	9	3	1	4	0	7
1	6	0	5	2	6	9	3	1	4	8	7
2	2	0	1	2	6	9	3	5	4	8	7
3	5	0	1	2	6	9	3	5	4	8	7
4	7	0	1	2	3	9	6	5	4	8	7
5	6	0	1	2	3	4	6	5	9	8	7

# Selection Sort

i	min	0	1	2	3	4	5	6	7	8	9
		8	5	2	6	9	3	1	4	0	7
0	8	8	5	2	6	9	3	1	4	0	7
1	6	0	5	2	6	9	3	1	4	8	7
2	2	0	1	2	6	9	3	5	4	8	7
3	5	0	1	2	6	9	3	5	4	8	7
4	7	0	1	2	3	9	6	5	4	8	7
5	6	0	1	2	3	4	6	5	9	8	7
6	6	0	1	2	3	4	5	6	9	8	7



# Selection Sort

i	min	0	1	2	3	4	5	6	7	8	9
		8	5	2	6	9	3	1	4	0	7
0	8	8	5	2	6	9	3	1	4	0	7
1	6	0	5	2	6	9	3	1	4	8	7
2	2	0	1	2	6	9	3	5	4	8	7
3	5	0	1	2	6	9	3	5	4	8	7
4	7	0	1	2	3	9	6	5	4	8	7
5	6	0	1	2	3	4	6	5	9	8	7
6	6	0	1	2	3	4	5	6	9	8	7
7	9	0	1	2	3	4	5	6	9	8	7

# Selection Sort

i	min	0	1	2	3	4	5	6	7	8	9
		8	5	2	6	9	3	1	4	0	7
0	8	8	5	2	6	9	3	1	4	0	7
1	6	0	5	2	6	9	3	1	4	8	7
2	2	0	1	2	6	9	3	5	4	8	7
3	5	0	1	2	6	9	3	5	4	8	7
4	7	0	1	2	3	9	6	5	4	8	7
5	6	0	1	2	3	4	6	5	9	8	7
6	6	0	1	2	3	4	5	6	9	8	7
7	9	0	1	2	3	4	5	6	9	8	7
8	8	0	1	2	3	4	5	6	7	8	9

# Selection Sort

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

# Bubble Sort

- Dizi elemanlarını baştan sona iki iki karşılaştır, büyük olanı sona taşı.
- En büyük eleman kabarcık gibi dizinin en sonuna doğru taşınır.

**for**  $i \leftarrow 0$  **to**  $n - 2$  **do**

**for**  $j \leftarrow 0$  **to**  $n - 2 - i$  **do**

**if**  $A[j + 1] < A[j]$

$\text{tmp} \leftarrow A[j]$

$A[j] \leftarrow A[j + 1]$

$A[j + 1] \leftarrow \text{tmp}$

- $O(n^2)$

# Bubble Sort $i=0$

0	1	2	3	4	5	6	7	8	9
8	5	2	6	9	3	1	4	0	7
8	5	2	6	9	3	1	4	0	7

# Bubble Sort $i=0$

0	1	2	3	4	5	6	7	8	9
8	5	2	6	9	3	1	4	0	7
8	5	2	6	9	3	1	4	0	7
5	8	2	6	9	3	1	4	0	7

# Bubble Sort $i=0$

0	1	2	3	4	5	6	7	8	9
8	5	2	6	9	3	1	4	0	7
8	5	2	6	9	3	1	4	0	7
5	8	2	6	9	3	1	4	0	7
5	2	8	6	9	3	1	4	0	7

# Bubble Sort $i=0$

0	1	2	3	4	5	6	7	8	9
8	5	2	6	9	3	1	4	0	7
8	5	2	6	9	3	1	4	0	7
5	8	2	6	9	3	1	4	0	7
5	2	8	6	9	3	1	4	0	7
5	2	6	8	9	3	1	4	0	7



# Bubble Sort $i=0$

0	1	2	3	4	5	6	7	8	9
8	5	2	6	9	3	1	4	0	7
8	5	2	6	9	3	1	4	0	7
5	8	2	6	9	3	1	4	0	7
5	2	8	6	9	3	1	4	0	7
5	2	6	8	9	3	1	4	0	7
5	2	6	8	9	3	1	4	0	7

# Bubble Sort $i=0$

0	1	2	3	4	5	6	7	8	9
8	5	2	6	9	3	1	4	0	7
8	5	2	6	9	3	1	4	0	7
5	8	2	6	9	3	1	4	0	7
5	2	8	6	9	3	1	4	0	7
5	2	6	8	9	3	1	4	0	7
5	2	6	8	9	3	1	4	0	7
5	2	6	8	3	9	1	4	0	7

# Bubble Sort $i=0$

0	1	2	3	4	5	6	7	8	9
8	5	2	6	9	3	1	4	0	7
8	5	2	6	9	3	1	4	0	7
5	8	2	6	9	3	1	4	0	7
5	2	8	6	9	3	1	4	0	7
5	2	6	8	9	3	1	4	0	7
5	2	6	8	9	3	1	4	0	7
5	2	6	8	3	9	1	4	0	7
5	2	6	8	3	1	9	4	0	7

# Bubble Sort $i=0$

0	1	2	3	4	5	6	7	8	9
8	5	2	6	9	3	1	4	0	7
8	5	2	6	9	3	1	4	0	7
5	8	2	6	9	3	1	4	0	7
5	2	8	6	9	3	1	4	0	7
5	2	6	8	9	3	1	4	0	7
5	2	6	8	9	3	1	4	0	7
5	2	6	8	3	9	1	4	0	7
5	2	6	8	3	1	9	4	0	7
5	2	6	6	3	1	4	9	0	7

# Bubble Sort $i=0$

0	1	2	3	4	5	6	7	8	9
8	5	2	6	9	3	1	4	0	7
8	5	2	6	9	3	1	4	0	7
5	8	2	6	9	3	1	4	0	7
5	2	8	6	9	3	1	4	0	7
5	2	6	8	9	3	1	4	0	7
5	2	6	8	9	3	1	4	0	7
5	2	6	8	3	9	1	4	0	7
5	2	6	8	3	1	9	4	0	7
5	2	6	8	3	1	4	9	0	7
5	2	6	8	3	1	4	0	9	7

# Bubble Sort $i=0$

0	1	2	3	4	5	6	7	8	9
8	5	2	6	9	3	1	4	0	7
8	5	2	6	9	3	1	4	0	7
5	8	2	6	9	3	1	4	0	7
5	2	8	6	9	3	1	4	0	7
5	2	6	8	9	3	1	4	0	7
5	2	6	8	9	3	1	4	0	7
5	2	6	8	3	9	1	4	0	7
5	2	6	8	3	1	9	4	0	7
5	2	6	8	3	1	4	9	0	7
5	2	6	8	3	1	4	0	9	7
5	2	6	8	3	1	4	0	7	9

# Bubble Sort $i=1$

0	1	2	3	4	5	6	7	8	9
5	2	6	8	3	1	4	0	7	9

# Bubble Sort $i=1$

0	1	2	3	4	5	6	7	8	9
5	2	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9



# Bubble Sort $i=1$

0	1	2	3	4	5	6	7	8	9
5	2	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9

# Bubble Sort $i=1$

0	1	2	3	4	5	6	7	8	9
5	2	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9

# Bubble Sort $i=1$

0	1	2	3	4	5	6	7	8	9
5	2	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	3	8	1	4	0	7	9

# Bubble Sort $i=1$

0	1	2	3	4	5	6	7	8	9
5	2	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	3	8	1	4	0	7	9
2	5	6	3	1	8	4	0	7	9
2	5	6	3	1	4	8	0	7	9

# Bubble Sort $i=1$

0	1	2	3	4	5	6	7	8	9
5	2	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	3	8	1	4	0	7	9
2	5	6	3	1	8	4	0	7	9
2	5	6	3	1	4	8	0	7	9
2	5	6	3	1	4	0	8	7	9

# Bubble Sort $i=1$

0	1	2	3	4	5	6	7	8	9
5	2	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	8	3	1	4	0	7	9
2	5	6	3	8	1	4	0	7	9
2	5	6	3	1	8	4	0	7	9
2	5	6	3	1	4	8	0	7	9
2	5	6	3	1	4	0	8	7	9
2	5	6	3	1	4	0	7	8	9

# Bubble Sort $i=2$

0	1	2	3	4	5	6	7	8	9
2	5	6	3	1	4	0	7	8	9

# Bubble Sort $i=2$

0	1	2	3	4	5	6	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	6	3	1	4	0	7	8	9



# Bubble Sort $i=2$

0	1	2	3	4	5	6	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	6	3	1	4	0	7	8	9

# Bubble Sort $i=2$

0	1	2	3	4	5	6	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	3	6	1	4	0	7	8	9

# Bubble Sort $i=2$

0	1	2	3	4	5	6	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	3	6	1	4	0	7	8	9
2	5	3	1	6	4	0	7	8	9

# Bubble Sort $i=2$

0	1	2	3	4	5	6	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	3	6	1	4	0	7	8	9
2	5	3	1	6	4	0	7	8	9
2	5	3	1	4	6	0	7	8	9

# Bubble Sort $i=2$

0	1	2	3	4	5	6	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	3	6	1	4	0	7	8	9
2	5	3	1	6	4	0	7	8	9
2	5	3	1	4	6	0	7	8	9
2	5	3	1	4	0	6	7	8	9

# Bubble Sort $i=2$

0	1	2	3	4	5	6	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	6	3	1	4	0	7	8	9
2	5	3	6	1	4	0	7	8	9
2	5	3	1	6	4	0	7	8	9
2	5	3	1	4	6	0	7	8	9
2	5	3	1	4	0	6	7	8	9
2	5	3	1	4	0	6	7	8	9

# Bubble Sort $i=3$

0	1	2	3	4	5	6	7	8	9
2	5	3	1	4	0	6	7	8	9

# Bubble Sort $i=3$

0	1	2	3	4	5	6	7	8	9
2	5	3	1	4	0	6	7	8	9
2	5	3	1	4	0	6	7	8	9



# Bubble Sort $i=3$

0	1	2	3	4	5	6	7	8	9
2	5	3	1	4	0	6	7	8	9
2	5	3	1	4	0	6	7	8	9
2	3	5	1	4	0	6	7	8	9

# Bubble Sort $i=3$

0	1	2	3	4	5	6	7	8	9
2	5	3	1	4	0	6	7	8	9
2	5	3	1	4	0	6	7	8	9
2	3	5	1	4	0	6	7	8	9
2	3	1	5	4	0	6	7	8	9

# Bubble Sort $i=3$

0	1	2	3	4	5	6	7	8	9
2	5	3	1	4	0	6	7	8	9
2	5	3	1	4	0	6	7	8	9
2	3	5	1	4	0	6	7	8	9
2	3	1	5	4	0	6	7	8	9
2	3	1	4	5	0	6	7	8	9

# Bubble Sort $i=3$

0	1	2	3	4	5	6	7	8	9
2	5	3	1	4	0	6	7	8	9
2	5	3	1	4	0	6	7	8	9
2	3	5	1	4	0	6	7	8	9
2	3	1	5	4	0	6	7	8	9
2	3	1	4	5	0	6	7	8	9
2	3	1	4	0	5	6	7	8	9

# Bubble Sort $i=3$

0	1	2	3	4	5	6	7	8	9
2	5	3	1	4	0	6	7	8	9
2	5	3	1	4	0	6	7	8	9
2	3	5	1	4	0	6	7	8	9
2	3	1	5	4	0	6	7	8	9
2	3	1	4	5	0	6	7	8	9
2	3	1	4	0	5	6	7	8	9
2	3	1	4	0	5	6	7	8	9

# Bubble Sort $i=4$

0	1	2	3	4	5	6	7	8	9
2	3	1	4	0	5	6	7	8	9

# Bubble Sort $i=4$

0	1	2	3	4	5	6	7	8	9
2	3	1	4	0	5	6	7	8	9
2	3	1	4	0	5	6	7	8	9

# Bubble Sort $i=4$

0	1	2	3	4	5	6	7	8	9
2	3	1	4	0	5	6	7	8	9
2	3	1	4	0	5	6	7	8	9
2	1	3	4	0	5	6	7	8	9



# Bubble Sort $i=4$

0	1	2	3	4	5	6	7	8	9
2	3	1	4	0	5	6	7	8	9
2	3	1	4	0	5	6	7	8	9
2	1	3	4	0	5	6	7	8	9
2	1	3	4	0	5	6	7	8	9

# Bubble Sort $i=4$

0	1	2	3	4	5	6	7	8	9
2	3	1	4	0	5	6	7	8	9
2	3	1	4	0	5	6	7	8	9
2	1	3	4	0	5	6	7	8	9
2	1	3	4	0	5	6	7	8	9
2	1	3	0	4	5	6	7	8	9

# Bubble Sort $i=4$

0	1	2	3	4	5	6	7	8	9
2	3	1	4	0	5	6	7	8	9
2	3	1	4	0	5	6	7	8	9
2	1	3	4	0	5	6	7	8	9
2	1	3	4	0	5	6	7	8	9
2	1	3	0	4	5	6	7	8	9
2	1	3	0	4	5	6	7	8	9

# Bubble Sort $i=5$

0	1	2	3	4	5	6	7	8	9
2	1	3	0	4	5	6	7	8	9

# Bubble Sort $i=5$

0	1	2	3	4	5	6	7	8	9
2	1	3	0	4	5	6	7	8	9
1	2	3	0	4	5	6	7	8	9

# Bubble Sort $i=5$

0	1	2	3	4	5	6	7	8	9
2	1	3	0	4	5	6	7	8	9
1	2	3	0	4	5	6	7	8	9
1	2	3	0	4	5	6	7	8	9

# Bubble Sort $i=5$

0	1	2	3	4	5	6	7	8	9
2	1	3	0	4	5	6	7	8	9
1	2	3	0	4	5	6	7	8	9
1	2	3	0	4	5	6	7	8	9
1	2	0	3	4	5	6	7	8	9

# Bubble Sort $i=5$

0	1	2	3	4	5	6	7	8	9
2	1	3	0	4	5	6	7	8	9
1	2	3	0	4	5	6	7	8	9
1	2	3	0	4	5	6	7	8	9
1	2	0	3	4	5	6	7	8	9
1	2	0	3	4	5	6	7	8	9



# Bubble Sort $i=6$

0	1	2	3	4	5	6	7	8	9
1	2	0	3	4	5	6	7	8	9

# Bubble Sort $i=6$

0	1	2	3	4	5	6	7	8	9
1	2	0	3	4	5	6	7	8	9
1	2	0	3	4	5	6	7	8	9

# Bubble Sort $i=6$

0	1	2	3	4	5	6	7	8	9
1	2	0	3	4	5	6	7	8	9
1	2	0	3	4	5	6	7	8	9
1	0	2	3	4	5	6	7	8	9

# Bubble Sort $i=6$

0	1	2	3	4	5	6	7	8	9
1	2	0	3	4	5	6	7	8	9
1	2	0	3	4	5	6	7	8	9
1	0	2	3	4	5	6	7	8	9
1	0	2	3	4	5	6	7	8	9

# Bubble Sort $i=6$

0	1	2	3	4	5	6	7	8	9
1	2	0	3	4	5	6	7	8	9
1	2	0	3	4	5	6	7	8	9
1	0	2	3	4	5	6	7	8	9
1	0	2	3	4	5	6	7	8	9

# Bubble Sort $i=7$

0	1	2	3	4	5	6	7	8	9
1	0	2	3	4	5	6	7	8	9

# Bubble Sort $i=7$

0	1	2	3	4	5	6	7	8	9
1	0	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9

# Bubble Sort $i=7$

0	1	2	3	4	5	6	7	8	9
1	0	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9



# Bubble Sort $i=8$

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9

# Bubble Sort $i=8$

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9

# Insertion Sort

- Eklemeli Sıralama (Insertion Sort)
- İskambil kağıtlarını elimizde sıralar gibi bir algoritma
- Sıradan bir yeni eleman al, önceki sıralı kısım ile karşılaştırarak olması gereken yere taşı
- **Kaç karşılaştırma?  $\sim N^2/2$**
- **Kaç yer değiştirme?  $\sim N^2/2$**



# Insertion Sort PseudoCode

INSERTION\_SORT (A,n)

1. for  $i \leftarrow 1$  to  $n-1$
2.      $\text{key} \leftarrow A[i]$
3.     //  $A[i]$ 'yi sıralı  $A[1..i-1]$  dizisine ekle
4.      $j \leftarrow i - 1$
5.     while  $j > 0$  and  $A[j] > \text{key}$
6.          $A[j+1] \leftarrow A[j]$
7.          $j \leftarrow j - 1$
8.      $A[j+1] \leftarrow \text{key}$

# Insertion Sort

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

# Insertion Sort

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

# Insertion Sort

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

# Insertion Sort

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



# Insertion Sort

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

# Insertion Sort

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

# Insertion Sort

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

# Insertion Sort

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

# Insertion Sort

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

# Insertion Sort

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

# Insertion Sort

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

# Insertion Sort

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



# Insertion Sort

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