



Sıralama

Aynı türden nesnelerden oluşan bir koleksiyonu, birbirini büyüklük-küçüklük açısından mantıksal bir düzene sokmak.

Girdi: $\{a_1, a_2, \dots, a_n\}$

Çıktı: $\{a'_1, a'_2, \dots, a'_n\}$ *permütasyon*; 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

Kullanım alanları:

- Transaction processing, kombinatoriyal optimizasyon, astrofizik, moleküler dinamik, dilbilim, genbilim, hava tahmini, ...

20.yy en iyi 10 algoritmasından biri: QuickSort !

Windows'u Etkinleştir

Windows'u etkinleştirmek için Ayarlar'a gidin.

00:08:19 / 01:38:20

2.9x



Sıralama Algoritmalarının Değerlendirilmesi

Performans:

- Çalışma zamanı
- Karşılaştırma ve değiştirme sayısı (değiştirme yoksa dizi erişim sayısı)

Ekstra Bellek:

- Yerinde sıralama (In-place) (Küçük bir bonksiyon çağrı 7'liği ya da sabit sayıda değişken olabilir): Raftaki kitapları rafta sıralama
- Harici bellek alanına sıralama (out of place sort): Raftaki kitapları yere döküp sıralama

Tekrar Eden Değerler:

- İstikrarlı (Stable): Girdide tekrar eden değerler, çıktıda aynı sıra ile yer alırlar. 1. tekrar ilk, 2. tekrar ikinci, N. tekrar sonuncu ...
- İstikrarsız (Unstable): Tekrarların yer garantisi yok.

Windows'u Etkinleştir

Windows'u etkinleştirmek için Ayarlar'a gidin.

00:13:25 / 01:38:20

2.9x



Temel Yöntemler

Selection Sort

Bubble Sort

Shell Sort

...

Windows'u Etkinleştir

Windows'u etkinleştirmek için Ayarlar'a gidin.

00:18:37 / 01:38:20

2.9x

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

tmp $\leftarrow A[j]$

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

$A[j + 1] \leftarrow tmp$

Windows'u Etkinleştir

Windows'u etkinleştirmek için Ayarlar'a gidin.

00:25:25 / 01:38:20

2.9x

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 ← 0 to n - 2 do
  for j ← 0 to n - 2 - i do
    if A[j + 1] < A[j]
      tmp ← A[j]
      A[j] ← A[j + 1]
      A[j + 1] ← tmp
```

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

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

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

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

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

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

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

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

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

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

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

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

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

00:37:11 / 01:38:30

2.8x

Selection Sort

.

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

00:49:28 / 01:38:30

2.8x

Selection Sort

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

.

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

00:55:49 / 01:38:30

2.8x

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.

*

Windows'u Etkinleştir

Windows'u etkinleştirmek için Ayarlar'a gidin.

00:57:07 / 01:38:20

2.9x

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
```

Windows'u Etkinleştir

Windows'u etkinleştirmek için Ayarlar'a gidin.

00:57:10 / 01:38:20

2.9x

Selection Sort

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

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

01:03:08 / 01:38:20

2.8x

