



Algoritmalar $\begin{cases} \text{Analiz} \\ \text{Tasarım} \end{cases}$

Soru: Performanstan daha önemli ne var?
Güvenlik, kullanıcı-dostluğu, doğruluk, modülerlik, programcı zamanı, basitlik

Sıralama Problemi

Girdi: Sayı dizisi $\langle a_1, a_2, \dots, a_n \rangle$

Çıktı: Sayı dizisi $\langle a'_1, a'_2, \dots, a'_n \rangle$
öyleki $a'_1 \leq a'_2 \leq \dots \leq a'_n$

Insertion (Araya Yerleştirme) Sıralaması

pseudocode
sözde kod
yalancı kod

```

Insertion_Sort(A, n) // A[1...n] i sıralar
for j=2 to n
    key = A[j]
    i = j-1
    while i > 0 and A[i] > key
        A[i+1] = A[i]
        i = i-1
    A[i+1] = key
  
```

indentation: girinti, hizalama

Değişmez: A:



1. Girdinin boyutuna bağlı: 100 ile 10^9 girdi boyutuna göre parametrize edilecek
2. Girdinin kendisine bağlı (örn: girdi sıralı ise)

2 4 5 8 10 12 17 en hızlı (iyi)
Ters Sıralı Olmalı

12 10 8 7 4 3 2 en yavaş (kötü)

Analiz Geçitleri

1. En kötü durum (worst-case) genellikle
 $T(n) = n$ boyutlu girdi için algoritmanın maximum çalışma zamanı
2. Ortalama durum (average case) bazen
 $T(n) = n$ boyutlu girdi için algoritmanın beklenen çalışma zamanı
3. En iyi durum (best case) nadiren

Asimptotik Analiz

$n \rightarrow \infty$ iken $T(n)$ in büyümesine bakarız.

Asimptotik Notasyon

Θ notasyonu: düşük dereceli terimleri sil
Baş katsayıyı ihmal et.

Örnekler: a. $T(n) = 3n^3 + 90n^2 - 50n + 6014$

$T(n) = \frac{\quad}{\quad}$

b. $T(n) = 3000n^2 + 100n \lg n$
 $T(n) = \frac{\quad}{\quad}$
 $\Theta(n^2)$

b algoritması a algoritmasından daha hızlı

$n=10$ için a daha hızlı olabilir.

... analiz bakıyoruz. yani

Araya Yerleştirme Algoritmasının Analizi

En iyi durum: → sabit zaman

$$T(n) = \sum_{j=2}^n \Theta(1) = \Theta\left(\sum_{j=2}^n 1\right)$$

$$= \Theta(n-1) = \Theta(n)$$

```

1: for j = 2 to A.length do
2:   key = A[j] Θ(1)
3:   i = j - 1 Θ(1)
4:   while i > 0 and A[i] > key do
5:     A[i+1] = A[i]
6:     i = i - 1
7:   end while
8:   A[i+1] = key Θ(1)
9: end for

```

En kötü durum:

$$T(n) = \sum_{j=2}^n \Theta(j) = \Theta\left(\sum_{j=2}^n j\right)$$

$$= \Theta\left(\frac{n(n+1)}{2} - 1\right)$$

$$= \Theta\left(\frac{n^2}{2} + \frac{n}{2} - 1\right)$$

$$= \Theta(n^2)$$

```

1: for j = 2 to A.length do
2:   key = A[j]
3:   i = j - 1
4:   while i > 0 and A[i] > key do ✓
5:     A[i+1] = A[i]
6:     i = i - 1
7:   end while
8:   A[i+1] = key
9: end for

```