

# Algoritma Analizi

Karmaşıklık Sınıfları  
(Complexity Classes)

# Algoritma Analizi

- Bilgisayar bilimlerine matematikten miras kalan karmaşıklık teorisi (complexity theory) konularından birisidir. Aslında problem veya algoritmaların çözüme ulaşmadaki karmaşıklığını ölçmek için kullanılır.
- Bu tanımın ardında, her problem veya algoritmanın bir fonksiyon gibi düşünülebilmesi, ve matematikteki fonksiyonların karmaşıklığını sınıflandırmak için kullanılan karmaşıklık sınıflarının (complexity classes, asimptotik notasyonlar) , algoritmalar için uygulanması yatmaktadır.

# Karmaşıklık Sınıfları

- Bilgisayar bilimlerinde kullanılan karmaşıklık sınıfları 5 tanedir.
- Küçük-o (small-o)
- Büyük-O (big-o, veya big-oh diye de geçer)
- Teta (Theta  $\Theta$ , sadece büyük tetadan bahsedebiliriz)
- Büyük omega (big- $\Omega$  )
- Küçük omega (small- $\omega$  )

# Karmaşıklık Sınıfları

- Bu beş sınıf, aynı fonksiyon için uygulanan ve bir fonksiyonun farklı özelliklerini anlatan sınıftır.
- Bu sınıflar, bir algoritma hakkında bilgi edinmemizi sağlar. Bu bilgileri aşağıdaki şekilde sıralayabiliriz:
- Bir algoritmanın ölçeği (Scalability)
- Bir algoritmanın zaman ve hafıza ihtiyacı (hızı ve ne kadar yer kaplayacağı) (time and memory efficiency)
- Bu sonuçlara, algoritmanın kullanıldığı giriş bilgisine göre ulaşılır. Bir algoritmanın işlediği veri miktarına göre ne kadar zaman ve ne kadar yer gerektiğini gösterir.

# Örnek

- Örneğin klasik bir arama algoritması olan doğrusal arama (linear search) algoritmasını ele alalım.
- Bu algoritma verilen bir dizide karışık olarak duran sayılardan bir tanesini bulmayı hedefler.
- `int a[10] = { 3, 7, 2, 5, 6, 1, 8, -12, 11, 45};`
- Yukarıdaki dizide bulunan bir sayının kaçinci sırada olduğunu aramak isteyelim. Örneğin 8 kaçinci sıradaki sayıdır?

# Örnek

- Algoritma, ilk elemandan başlayarak son elemana kadar bütün sayılara bakar ve aranan sayıya rastlarsa sonlanır.
- Bu durumda aranan 8 sayısını bulmak için sırasıyla 3,7,2,5,6 ve 1 sayılarına bakılacak, nihayet 8 sayısı bulununca 7. sırada bulunduğu sonucu ile algoritma bitecektir.

# Örnek

- Algoritmanın yer ihtiyacı ne kadardır?
- Bu algoritmanın üzerinde çalıştığı veri büyüklüğü dizinin büyüklüğüdür ve bu değer 10'dur.
- Dolayısıyla arama işleminin  $n$  sayı içerisinde yapılacağını düşünürsek,  $n$  sayıyı hafızada tutmamız gerektiğini görürüz.
- Demek ki algoritmamızın hafıza karmaşıklığı (memory complexity)  $n$ 'dir.

# Örnek

- Algoritmanın çalışma hızı nedir?
- Arama işleminin  $n$  adet sayı üzerinde yapıldığını söylersek, aşağıdaki 3 durumu incelememiz gerekir:
  - En kötü ihtimalle kaç adımda bulunur? (worst case)
  - En iyi ihtimalle kaç adımda bulunur? (best case)
  - Ortalama kaç adımda bulunur? (average case)



# Örnek

- Bu ihtimal analizini doğrusal arama algoritmamıza uyguladığımızda, karşılaşılabileceğimiz en kötü durum aranan sayının dizinin en sonunda bulunması veya dizide hiç bulunmamasıdır.
- Bu durumda dizideki bütün elemanlara bakılması gerekecektir. Dolayısıyla en kötü durumda karmaşıklığımız  $n$  sayı için  $n$  olacaktır.
- En iyi ihtimal ise, ilk bakılan sayının, aranan sayı olmasıdır. Bu durumda tek bir bakma işlemi yeterlidir. Bu durumdaki karmaşıklığımız ise 1 olacaktır.

# Örnek

- Ortalama durum ise bu algoritmanın çok sefer çalışması sonucunda istatistiksel olarak ortalama kaç elemana bakılacağıdır.
- Bu dizide bulunan sayıların hepsinin aranma oranlarının eşit olduğunu kabul edersek, ortalama durum  $n/2$  olur.
- İşte yukarıdaki bu durum analizleri, bizim karmaşıklık sınıflarımızı veren analizlerdir.
- $O$  – En kötü durum analizi
- $\Theta$  – Ortalama durum analizi
- $\Omega$  – En iyi durum analizi

# Örnek

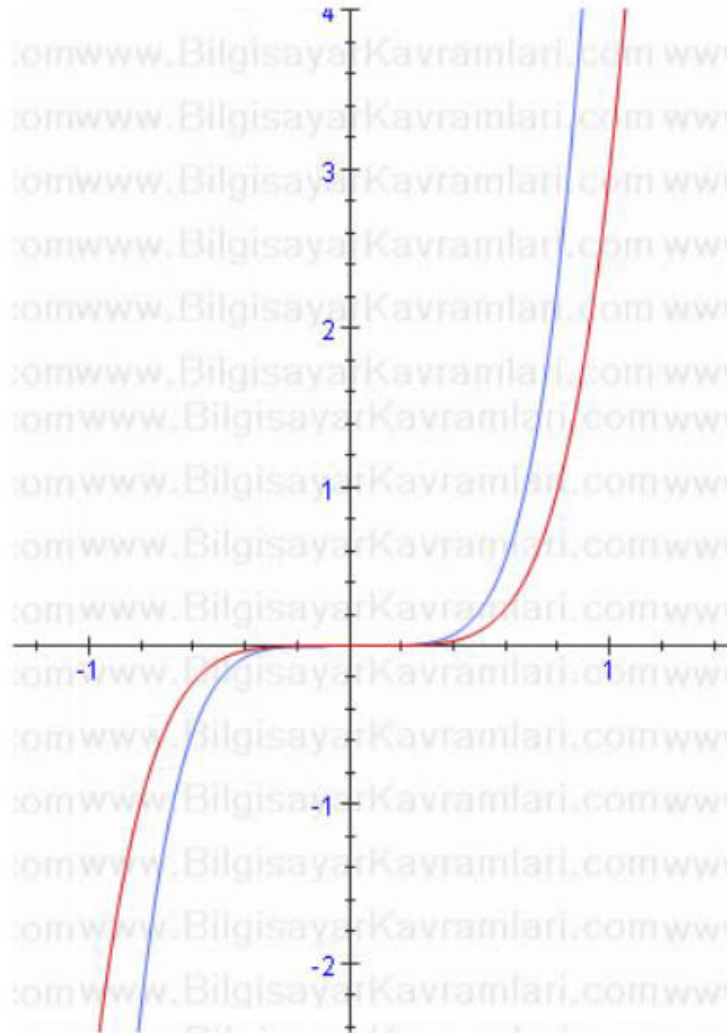
Aynı zamanda, bir fonksiyon için de o gösterimi üst sınırı (upper bound), gösterimi ise alt sınırı (lower bound) ifade eder.

Örnek olarak aşağıdaki fonksiyonu ele alacak olursak.

$$f(x) = 3x^5$$

Bu fonksiyona bir üst sınır çizmek istenirse, örneğin  $7x^5$  fonksiyonu çizilebilir.

# Örnek



# Örnek

Mavi fonksiyon  $7x^5$  ve kırmızı fonksiyon ise  $f(x)$  fonksiyonudur.

Yukarıdaki şekilde görüldüğü üzere,  $x=0$  noktasından büyük değerler için yani  $x>0$  için, mavi renkle gösterilen fonksiyon, kırmızı fonksiyonun üst sınırını oluşturmaktadır ve denilebilir ki, kırmızı fonksiyon hiçbir zaman mavi fonksiyonun üzerine geçemez.

Bu durumda mavi fonksiyon, kırmızının üst sınırı (upper bound) olmuş olur. Diğer bir deyişle, mavi fonksiyon, kırmızı fonksiyondan daha kötü bir fonksiyondur ve kırmızı fonksiyon en kötü ihtimalle mavi fonksiyon kadar kötü olabilir.

Bu durumu büyük-o ile gösterirsek:

$$f(x) \in O(x^5)$$

şeklinde yazmamız yeterlidir. Bu gösterimin açılımı aşağıda verilmiştir:

$f(x) \in O(g(x))$  , öyle bir  $c$  değeri bulunursa ki,  $x_0>0$  için,  $x>x_0$  şartıyla, bütün  $f(x) \leq c g(x)$  olsun.

# Örnek

Bu tanım, kısaca, şayet  $f(x) \in \Omega(g(x))$  bağlantısı verilirse, herhangi bir  $x_0$  noktasından başlanarak sonsuza kadar bütün  $f(x) \geq c g(x)$  koşulunu sağlayan bir  $c$  değerinin bulunabileceğini anlatmaktadır.

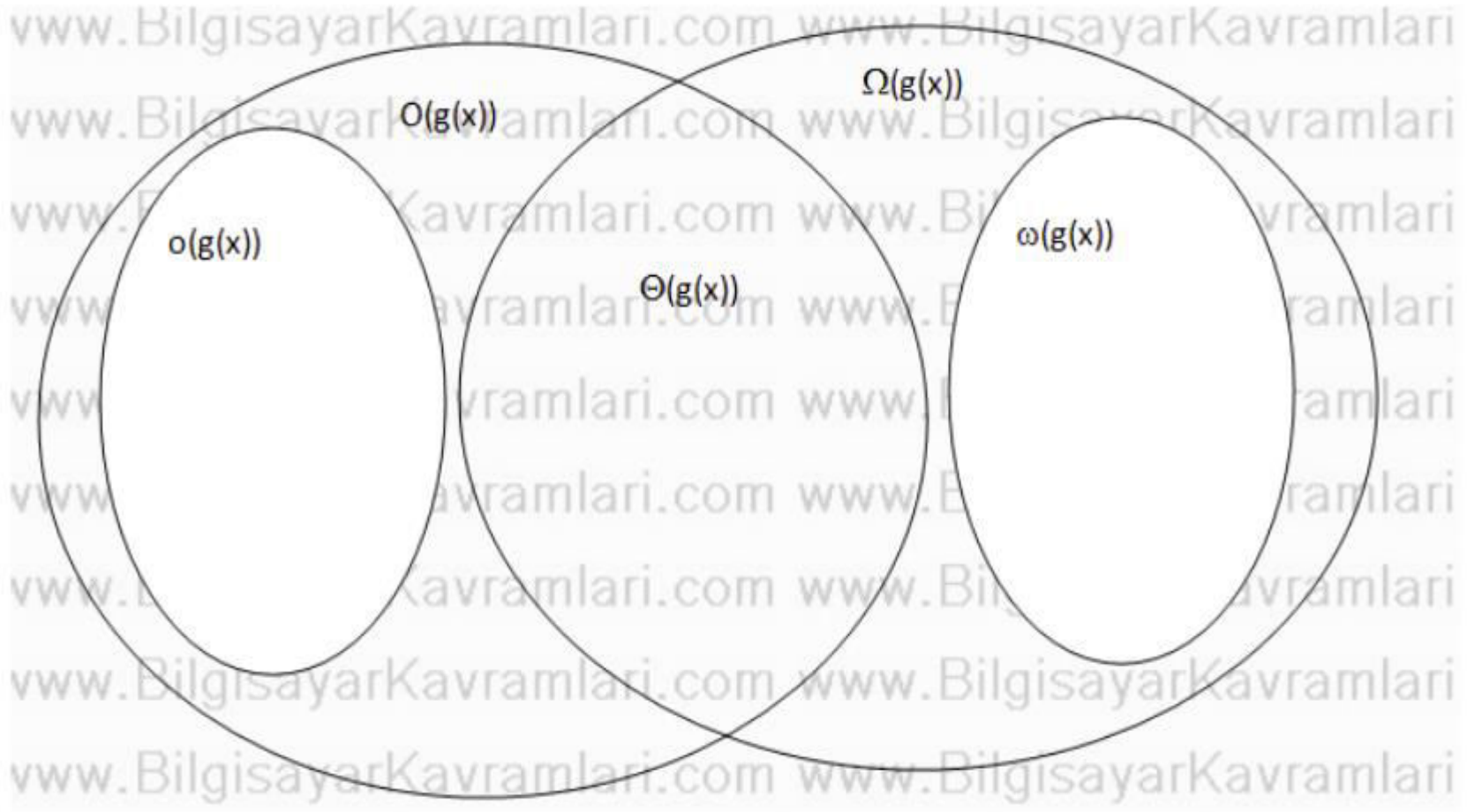
Bu iki tanımda dikkat edilecek bir nokta,  $f(x)$  ile  $c \cdot g(x)$  arasındaki bağlantının eşitliği de kapsamasıdır. Yani örneğin big-omega için,  $c \cdot g(x)$  değeri  $f(x)$  fonksiyonundan küçük veya eşit olabilir.

Şayet bu tanımdaki eşitliği kaldırırsak, küçük-o ve küçük- $\omega$  tanımını elde ederiz. Yani küçük-o için,  $c \cdot g(x)$  değerinin  $f(x)$  değerinden sürekli olarak büyük olması gerekir.

Theta gösterimi ise ortalama durum, yani eşitlik durumu içindir. Burada aranan şart  $f(x) \in \Theta(g(x))$  için  $c_1 \cdot g(x) \leq f(x) \leq c_2 \cdot g(x)$ 'dir. Yani  $g(x)$  fonksiyonunun  $f(x)$  fonksiyonu ile aynı şekilde hareket ettiğini göstereniki sabit sayı,  $c_1$  ve  $c_2$  bulunabiliyorsa  $f(x) \in \Theta(g(x))$  denilebilir.

Bu durumda yukarıda listelediğimiz 5 sınıf aşağıdaki şekilde ilişkilendirebiliriz:

# Örnek



# Örnek

- Yukarıdaki şekilde anlatılmak istenen, sınıflar arasında, bazı kesişim durumlarının olduğudur.
- Görüldüğü üzere, en kötü durum ( $\text{big-o}$ ) ile en iyi durum ( $\text{big-}\Omega$ ) kesişmesi sadece eşitlik durumunda olur.
- Bu durum ise teta ile gösterilmektedir.
- Benzer şekilde küçük-o ve küçük omega hiçbir zaman kesişemez ve büyük-o ile küçük-o arasında ve büyük omega ile küçük omega arasındaki fark alanında eşitlik durumları bulunur.