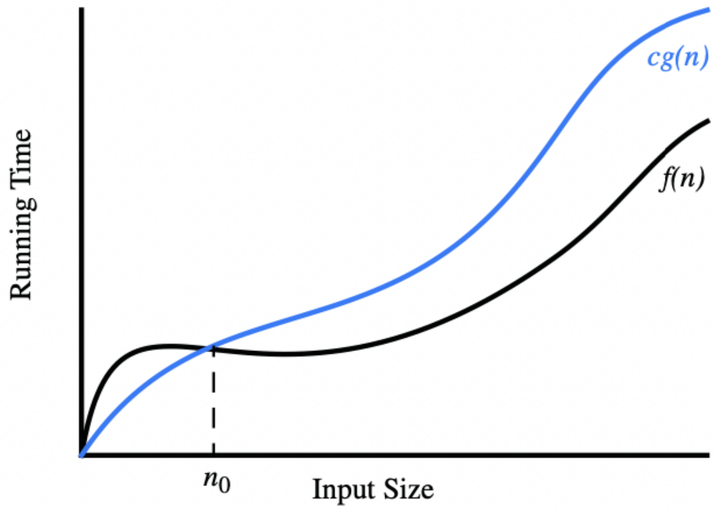
Big Oh

Matematiksel Tanım

Let $f(n)$ and $g(n)$ be functions mapping positive integers to positive real numbers. We say that $f(n)$ is $O(g(n))$ if there is a real constant $c > 0$ and an integer constant $n_0 \geq 1$ such that

$$f(n) \leq c \cdot g(n), \quad \text{for} \quad n \geq n_0.$$

This definition is often referred to as the "big-Oh" notation, for it is sometimes pronounced as "$f(n)$ is ***big-Oh*** of $g(n)$." Figure 4.5 illustrates the general definition.



Bu tanıma çok takılmanıza gerek yok. Ancak bütün notasyonun çıkış noktasının bu olduğunu bilmeniz yeterli.

Tanımdan Big-O notasyonuyla fonksiyona bir üst sınır getirdiğimizi görüyoruz. Eğer f = O(g) ise belli bir noktadan sonra f daima g'den daha yavaş büyüyecektir ve bu yüzden altında kalacaktır.

Big-O'nun Özellikleri

> *The big-Oh notation allows us to ignore constant factors and lower-order terms and focus on the main components of a function that affect its growth.*

Big-O notasyonunda sabit terimleri ve düşük dereceli terimleri göz ardı ederiz. Çünkü bunlar fonksiyonun büyümesini etkileyen temel faktörler değildir. Büyüklüğü belirleyen tek terim en büyük dereceli terimdir. Bunu polinomlar için şöyle genelleyebiliriz.

Proposition 4.8: *If $f(n)$ is a polynomial of degree d, that is,*

$$f(n) = a_0 + a_1 n + \cdots + a_d n^d,$$

and $a_d > 0$, then $f(n)$ is $O(n^d)$.

Fark ettiğiniz gibi katsayılar ve düşük dereceli terimler bu notasyonda önemli değil. Çünkü katsayıları ne olursa olsun, bir fonksiyonun diğerini belli bir süreden sonra geçip geçememesi sadece en yüksek dereceli terime bağlıdır. Şu örneğe bakalım:

$$f(n) = 100000 * n \qquad\qquad g(n) = n^2$$

f'in katsayısının ne kadar olduğu önemli değil. Belli bir değerden sonra g mutlaka f'i geçecektir çünkü $f(n) = O(n), g(n) = n^2$. Yani g, f'den daha hızlı büyüyor ve onu aşıyor. Zaten $n > 100000$ için böyle olduğunu rahatça görebiliriz.

Örnekler

Example 4.7: $5n^4 + 3n^3 + 2n^2 + 4n + 1$ *is* $O(n^4)$.

Justification: *Note that* $5n^4 + 3n^3 + 2n^2 + 4n + 1 \leq (5+3+2+4+1)n^4 = cn^4$, *for* $c = 15$, *when* $n \geq n_0 = 1$. ■

Example 4.9: $5n^2 + 3n\log n + 2n + 5$ *is* $O(n^2)$.

Justification: $5n^2 + 3n\log n + 2n + 5 \leq (5+3+2+5)n^2 = cn^2$, for $c = 15$, when $n \geq n_0 = 1$. ■

Example 4.10: $20n^3 + 10n\log n + 5$ *is* $O(n^3)$.

Justification: $20n^3 + 10n\log n + 5 \leq 35n^3$, for $n \geq 1$. ■

Example 4.11: $3\log n + 2$ *is* $O(\log n)$.

Justification: $3\log n + 2 \leq 5\log n$, for $n \geq 2$. Note that $\log n$ is zero for $n = 1$. That is why we use $n \geq n_0 = 2$ in this case. ■

Example 4.12: 2^{n+2} *is* $O(2^n)$.

Justification: $2^{n+2} = 2^n \cdot 2^2 = 4 \cdot 2^n$; hence, we can take $c = 4$ and $n_0 = 1$ in this case. ■

Example 4.13: $2n + 100\log n$ *is* $O(n)$.

Justification: $2n + 100\log n \leq 102n$, for $n \geq n_0 = 1$; hence, we can take $c = 102$ in this case. ■