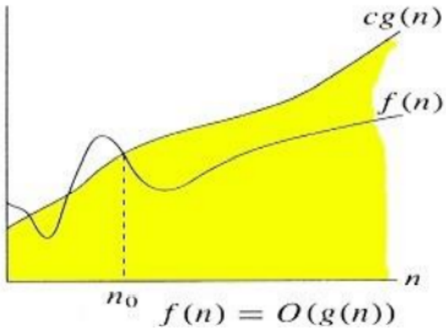
Asymptotic Notations:

It is often used to describe how the size of the input data affects an algorithm's usage of computational resources. Running time of an algorithm is described as a function of input size n for large n.

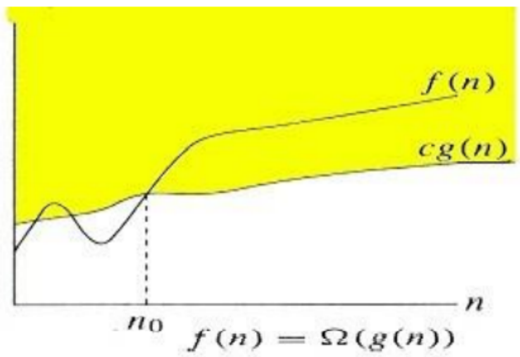
Big oh(O): Definition: f(n) = O(g(n)) (read as f of n is big oh of g of n) if there exist a positive integer n_0 and a positive number c such that $|f(n)| \leq c|g(n)|$ for all $n \geq n_0$. Here g(n) is the upper bound of the function f(n).

f(n)	g(n)	
$16n^3 + 45n^2 + 12n$	n^3	$f(n) = O(n^3)$
$34n - 40$	n	f(n) = O(n)
50	1	f(n) = O(1)



Omega(Ω): Definition: f(n) = Ω(g(n)) (read as f of n is omega of g of n), if there exists a positive integer n_0 and a positive number c such that $|f(n)| \geq c\ |g(n)|$ for all $n \geq n_0$. Here g(n) is the lower bound of the function f(n).

f(n)	g(n)	
$16n^3 + 8n^2 + 2$	n^3	$f(n) = \Omega(n^3)$
24n + 9	n	f(n) = Ω(n)



Theta(Θ): Definition: f(n) = Θ(g(n)) (read as f of n is theta of g of n), if there exists a positive integer n_0 and two positive constants c_1 and c_2 such that $c_1\ |g(n)| \leq |f(n)| \leq c_2\ |g(n)|$ for all $n \geq n_0$. The function g(n) is both an upper bound and a lower bound for the function f(n) for all values of n, $n \geq n_0$.

f(n)	g(n)	
$16n^3 + 30n^2 - 90$	n^2	$f(n) = \Theta(n^2)$
$7.2^n + 30n$	2^n	$f(n) = \Theta(2^n)$