Topics to discuss

1. Asymptotic Notation
2. Big-O Notation
3. Examples
4. Big-O Visualization.
5. Why it is called Asymptotic Notation.

Asymptotic Notation :

To compare two or more than two Algorithms rate of growth with respect to time and space we need asymptotic notation.

It is a tool to represent the time and space complexity of algorithms for asymptotic analysis.

Problem

A_1
10sec
10byte

A_2
2sec
5byte

A_3
5sec
20byte.

There are mainly three asymptotic notations:

1) Big-O Notation (O-notation)
2) Big Omega Notation (Ω-notation)
3) Theta Notation (Θ-notation)

Big-O Notation

Lets assume that a given algorithm is represented in the form of function $f(n)$.

Generally, it is represented as $f(n) = O(g(n))$.

This notation gives the tight upper bound of the given function.

That means, at larger value of n, the upper bound of $f(n)$ is $g(n)$.

Definition:

Big-O notation defined as $O(g(n)) = \{ f(n)$: there exist positive constants c and n_0 such that $0 \le f(n) \le c\, g(n)$ for all $n \ge n_0 \}$.

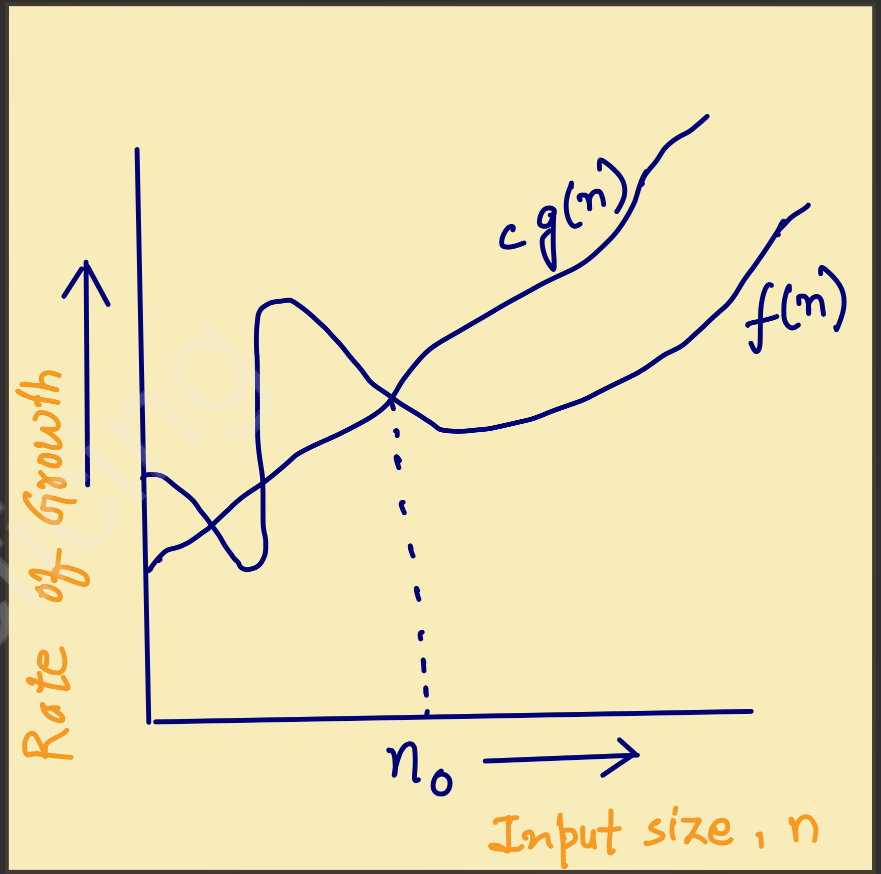
This simply means that f(n) does not grow faster than g(n).

g(n) is an asymptotic tight upper bound for f(n).

n_0 is the point from which we need to consider the rate of growth for a given algorithm.

Below n_0, the rate of growth could be different.

n_0 is called threshold for the given function



1. Find the upper bound for $f(n) = 2n+4$

Solution :- $f(n) = 2n+4$

By definition, $0 \leq f(n) \leq c \cdot g(n)$

$f(n) \leq c \cdot g(n)$

$2n+4 \leq 3n$

$c = 3$

$g(n) = n$

$2n+4-3n \leq 0$

$-n \leq -4$

$n \geq 4$

$f(n) = O(g(n))$

$f(n) = O(n)$

where

$c = 3$

$n \geq 4$

$n_0 = 4$

(2) Find the upper bound for $f(n) = n^2 + 1$

Solution :- $f(n) = n^2 + 1$

$f(n) \le c \cdot g(n)$

$n^2 + 1 \le 2n^2$

$c = 2$

$g(n) = n^2$

$n^2 + 1 - 2n^2 \le 0$

$-n^2 \le -1$

$n \ge 1$

$f(n) = O(g(n))$

$f(n) = O(n^2)$

for $c = 2$

$n \ge 1$

$n_0 = 1$

Uniqueness: There is no unique set of values of n_0 and c in proving the asymptotic bounds.

$f(n) = 100n + 5$ $\qquad$ $f(n) \leq c \cdot g(n)$

$100n + 5 \leq 100n + n$

$100n + 5 \leq 101n$

$c = 101$

$g(n) = n$

$100n + 5 - 101n \leq 0$

$-n \leq -5$

$n \geq 5$

$$\boxed{\begin{array}{c} f(n) = O(g(n)) \\ f(n) = O(n) \\ c = 101, \; n \geq 5, \; n_0 = 5 \end{array}}$$

$100n + 5 \leq 100n + 5n$

$100n + 5 \leq 105n$

$c = 105$

$g(n) = n$

$100n + 5 - 105n \leq 0$

$-5n \leq -5$

$n \geq 1$

$$\boxed{\begin{array}{c} f(n) = O(n) \\ c = 105, \; n \geq 1, \; n_0 = 1 \end{array}}$$

Big-O Visualization :

$O(g(n))$ is the set of functions with smaller or the same order of growth as $g(n)$.

For eg: $O(n^2)$ includes $O(1)$, $O(n)$, $O(n\log n)$ etc.

$O(1)$: 100, 1000, 2000

$O(n)$: $3n+100$, $10n$, 5

$O(n\log n)$: $5n\log n$, $3n$, 4

$O(n^2)$: n^2, $n-4$, $3n\log n$, 3

Why it is called Asymptotic Notation

In every case for a given function $f(n)$, we are trying to find another function $g(n)$ which approximates $f(n)$ at higher values of n. That means $g(n)$ is also a curve which approximates $f(n)$ at higher values of n.

In mathematics we call such a curve an asymptotic curve. In other terms, $g(n)$ is the asymptotic curve for $f(n)$.

For this reason, we call algorithm analysis as asymptotic analysis.

Follow Now

Start Practicing

i._am._arfin

Arfin Parween