COMSATS Sahiwal

Computer Science Department

Subject:

Data Structures and Algorithms

Submitted by:

M. Adnan Tariq
FA22-BCS-082

Submitted to:

Mam Tahreem Saeed

Asymptotic Notations

Big-Oh Notation

Big-Oh notation often denoted as $O(f(n))$ is a mathematical notation used in computer science to describe the upper bound or worst-case time complexity of an algorithm in terms of input size.

- "$f(n)$" represents a function, that characterizes the upper bound of an algorithm.
- "n" represent the size of input.

Example 01

Prove that running time $T(n) = n^3 + 20n + 1$ is $O(n^3)$

By Big-Oh definition

$T(n)$ is $O(n^3)$ if $T(n) \leq c \cdot n^3$

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

$$n^3 + 20n + 1 \leq c \cdot n^3$$

$$\left(\frac{n^3 + 20n + 1}{n^3} \right) \leq c$$

$$1 + \frac{20}{n^2} + \frac{1}{n^2} \leq c.$$

hence $f(n) \in O(g(n))$, $c \geq 22$, $n_0 = 1$

Example 02

Show that $f(x) = x^3 + 3x - 2$ is $O(x^3)$

$$f(n) = x^3 + 3x - 2, \quad g(n) = x^3$$

Now we have to show that

$$f(n) \in O(g(n)).$$

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

$$x^3 + 3x - 2 \leq c \cdot x^3$$

$$\frac{x^3 + 3x - 2}{x^3} \leq c$$

$$1 + \frac{3}{x^2} - \frac{2}{x} \leq c.$$

hence $f(n) \in O(g(n))$, $c = 2$, $x = 1$

Big-Omega Notation

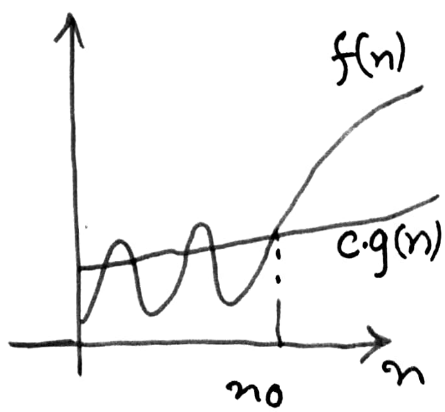
Big-Omega Notation, denoted as Ω is a mathematical notation used in computer science to describe the lower bound of growth rate of function or an algorithm.

- It provides information about best-case scenario for the performance of an algorithm.

- For a function $f(n)$ to be $\Omega(g(n))$, the following condition must hold;

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

$\therefore n \geq n_0$.



- "$f(n)$" is the function's actual performance
- "$g(n)$" is lower bound function
- "c" is positive constant.

Example 01

Show that the function
$f(x) = 3x^2 + 2x + 1$ is $\Omega(x^2)$.

$f(n) = 3x^2 + 2x + 1$, $g(n) = x^2$

Now we have to show that

$$f(n) \in \Omega(g(n)) ?$$

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

$$3x^2 + 2x + 1 \geq c(x^2)$$

$$\frac{3x^2 + 2x + 1}{x^2} \geq c.$$

$$3 + \frac{2}{x} + \frac{1}{x^2} \geq c.$$

Hence $f(n) \in \Omega(g(n))$ if
$c = 1$, $x = 2$.

Example 02

Show that $f(x) = x \in \Omega(x^2)$

$f(n) = x \quad , \quad g(n) = x^2$

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

$$x \geq c \cdot x^2$$

We could not find constants c, n that satisfy the condition for $f(x) = x$ being $\Omega(x^2)$.

Hence

$$f(x) \notin \Omega(g(x)).$$

Theta Notation

Big theta (Θ) notation is a mathematical notation commonly used in computer science to describe the tight or asymptotically optimal bound on the growth rate of a function/algorithm.

- It provides information about both the upper and lower bounds of growth rate of function, indicating that function's behaviour is bounded within a specific range.

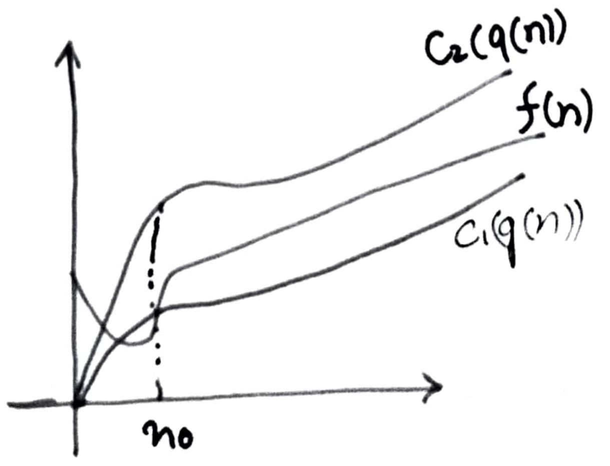
- For a function f(n) to be Θ(g(n)), the following must hold;

1- There exist positive constant c_1 and c_2 such that;

$$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n).$$
$$n \geq n_0.$$

2- The function $f(n)$ exhibits the same asymptotic growth rate as $g(n)$.

- Big Theta Notation is particularly useful for describing the average-case time complexity of algorithms.
- It indicates that the algorithm's performance is neither better nor worse than growth rate $g(n)$.



Example 01 :

Show that the function $f(x) = 4x^3 - 2x^2 + 7$ is $\theta(x^3)$

$f(x) = 4x^3 - 2x^2 + 7$, $g(n) = x^3$

$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$

$c_1 \cdot x^3 \leq 4x^3 - 2x^2 + 7 \leq c_2 x^3$

if we take $c_1 = 1$, $c_2 = 5$, $x = 1$ then this equation satisfies.

Hen $f(n) \in \theta(g(n))$, $c_1 = 1$, $c_2 = 5$, $x = 1$

Example 02

Show that $f(x) = x^2 \in \theta(x^3)$

$f(x) = x^2$, $g(n) = x^3$

$c_1 \cdot g(n) \leq f(x) \leq c_2 \cdot g(n)$

We could not find constants that can satisfy the equation.

Little-Oh Notation

Little-Oh notation denoted as "o", is a mathematical notation used in computer science to describe the relationship between the growth rates of two functions.

- It is used to denote that one function grows significantly faster than another as the input size becomes larger.

For a function f(n) to be o(g(n)), the following must hold;

$$\lim_{(n \to \infty)} \frac{f(n)}{g(n)} = 0$$

This means that the ratio of f(n) to g(n) approaches zero as

n becomes very large.

- o-Notation is used to denote a upper bound that is not asymptotically tight

Example:

Show that $f(x) = x^2 \in o(x^3)$

To prove that $f(x) = x^2 \in o(x^3)$, we need to show that the ratio $\frac{f(x)}{x^3}$ approaches to zero as x goes to infinity.

$$\lim_{(x \to \infty)} \frac{f(x)}{x^3} = \frac{x^2}{x^3} = \frac{1}{x} = 0$$

As x- approaches infinity, the limit becomes zero. This means that function $f(x) = x^2$ grows slower than x^3 as x becomes large.

little-Omega

little omega notation, denoted as "ω", is a mathematical notation used in computer science to describe relationship between the growth rates of two functions

- It is essentially the opposite of little-oh notation, indicating that one function grows strictly faster than another as the input size becomes larger.

For a function to be $\omega(g(n))$ the following must hold;

$$\lim_{(n \to \infty)} \frac{f(n)}{g(n)} = \infty$$

This means that ratio of $f(n)$ to $g(n)$ approaches infinity as n

becomes very large.

Example:

Show that $f(x) = x^3 \in \omega(x^2)$

Consider the limit

$$\lim_{x \to \infty} \frac{f(x)}{g(x)} = \frac{x^3}{x^2} = x = \infty$$

As x approaches infinity, the limit becomes infinite. This means that function $f(x) = x^3$ grows significantly faster than x^2 as x becomes very Large.