9.C.1 BIG THETA(Θ) NOTATION- MATHEMATICAL EXAMPLES AND PROOFS

EXAMPLE 1

Find Θ bound for $f(n) = \frac{n^2}{2} - \frac{n}{2}$

Solution

Note by definition:

$c_1 g(n) \le f(n) \le c_2 g(n) \;\; for\ all\ n \ge n_0$

Hence

$$\frac{n^2}{2} - \frac{n}{2} \le c_2 g(n)$$

$\Rightarrow f(n) \le \left(\frac{n^2}{2} + \frac{n^2}{2}\right) - \frac{n^2}{2}$ [As there is $-\frac{n}{2}$ $and\ \frac{2n^2}{2} \ge \frac{n^2}{2}$ $is\ 2$ $is\ degree\ of\ polynomial$]

$$\Rightarrow f(n) \le \left(\frac{n^2}{2} + \frac{n^2}{2}\right) - \frac{n^2}{2}$$

$$\Rightarrow f(n) \leq \left(\frac{2n^2}{2}\right) - \frac{n^2}{2}$$

$$\Rightarrow f(n) \leq n^2$$

Now let's take $\frac{n^2}{5}$ *which is* $\leq \frac{n^2}{2}$ i.e.

$$\Rightarrow \frac{n^2}{5} \leq \frac{n^2}{2} - \frac{n}{5} \leq n^2$$

When in such a condition we have to go each input as mentioned by the definition of big theta:

N	$\frac{n^2}{2} - \frac{n}{2}$	$\frac{n^2}{5}$	n^2
1	0	0.2	1
2	1	0.8	4
3	3	1.8	9
4	6	3.2	16

Here we cannot consider 1 as f(n) becomes 0 as it becomes.

$$0.2 \leq 0 \leq 1\text{ , }\mathit{here}\ 0.2 \leq 0\ \mathit{is\ not\ true}.$$

Hence, we say $n \geq 2$.

As we take $\frac{n^2}{5} \leq f(n), hence\ c_1 g(n) = \frac{1}{5} \times n^2$

Hence $c_1 = \frac{1}{5}$

Also take $f(n) \leq n^2, hence\ c_1 g(n) = 1 \times n^2$

Hence $c_2 = 1$

Also $n_0 = 2$.

Therefore $, f = \Theta(g)$

Hence, $f(n) = \Theta(g(n))$

$\Rightarrow \frac{n^2}{2} - \frac{n}{5} = \Theta(n^2)$

EXAMPLE 2

Prove $n \neq \Theta(n^2)$

Solution

Note by definition:

$c_1 g(n) \leq f(n) \leq c_2 g(n) \ \ for\ all\ n \geq n_0$

Here clearly mentions that $g(n) = n^2, f(n) = n$

Hence,

$c_1 \times n^2 \leq n \leq \ c_2 \times n^2$

$$\Rightarrow c_1 n^2 \leq n$$

$$\Rightarrow n^2 \leq \frac{n}{c_1}$$

$$\Rightarrow n \leq \frac{1}{c_1}$$

Also ,

$$\Rightarrow n \leq c_2 n^2$$

$$\Rightarrow \frac{n}{c_2} \leq n^2$$

$$\Rightarrow n \geq \frac{1}{c_2}$$

As it stands now:

$$\frac{1}{c_2} \le n \le \frac{1}{c_1}$$

And if we put according to definition:

$$\frac{1}{c_2} \times g(n) \le f(n) \le \frac{1}{c_1} \times g(n)$$

And $\Theta(n^2)$ *only can happen when it is in above condition. And it goes against the definition* of Big theta :

$$c_1 g(n) \le f(n) \le c_2 g(n) \;\; for\ all\ n \ge n_0$$

Hence, $n \ne \Theta(n^2)$.

EXAMPLE 3

$$Prove\ 6n^3 \ne \Theta(n^2)$$

Solution

Note by definition:

$$c_1 g(n) \le f(n) \le c_2 g(n) \;\; for\ all\ n \ge n_0$$

Here clearly mentions that $g(n) = n^2, f(n) = 6n^3$

Hence,

$c_1 n^2 \leq 6n^3 \leq c_2 n^2$

$\Rightarrow c_1 n^2 \leq 6n^3$

$\Rightarrow c_1 \leq \frac{6n^3}{n^2}$

$\Rightarrow c_1 \leq 6n$

$\Rightarrow \frac{c_1}{6} \leq n$

$\Rightarrow n \geq \frac{c_1}{6}$

Again,

$6n^3 \leq c_2 n^2$

$\Rightarrow 6n \leq c_2$

$\Rightarrow n \leq \frac{c_2}{6}$

As it stands now:

$$\frac{c_1}{6} \leq n \leq \frac{c_2}{6}$$

And if we put according to definition:

$$\frac{c_1}{6} \times g(n) \le f(n) \le \frac{c_2}{6} \times g(n)$$

And $\Theta(n^2)$ *only can happen when it is in above condition. And it goes against the definition* of Big theta :

$c_1 g(n) \le f(n) \le c_2 g(n)$ $for\ all\ n \ge n_0$

Hence, $6n^3 \neq \Theta(n^2)$.

EXAMPLE 4

$$Prove\ n \neq \Theta(logn)$$

Solution

Note by definition:

$c_1 g(n) \le f(n) \le c_2 g(n)$ $for\ all\ n \ge n_0$

Here clearly mentions that $g(n) = logn, f(n) = n$

$\therefore c_1 logn \le n \le c_2 logn$, $for\ all\ n \ge n_0$

$\therefore c_1 \le \frac{n}{logn}$ and $c_2 \ge \frac{n}{logn}$, $for\ all\ n \ge n_0$

If we observe:

$$\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = \lim_{n\to\infty}\left(\frac{n}{log(n)}\right) = \frac{\lim_{n\to\infty}(n)}{\lim_{n\to\infty}(logn)} = \frac{\infty}{\infty}$$

Applying L'Hospital Theorem: $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = \lim_{n\to\infty}\left(\frac{f'(n)}{g'(n)}\right)$

$$As\ we\ know\ that, \frac{d}{dn}(n) = 1\ and\ \frac{d}{dn}(log(n)) = \frac{1}{n}$$

Hence $\lim_{n\to\infty}\left(\frac{1}{\frac{1}{n}}\right) = \lim_{n\to\infty}(n) = \infty$

Hence there is no such finite constant for f(n)=n for which

$f(n) \leq c_2 \times g(n)$

Hence it is impossible to have $\Theta(logn)$ for f(n)=n.

EXAMPLE 5

Prove $5n^2 + 3n + 1 = \theta(n^2)$

Solution:

Let, $f(n) = 5n^2 + 3n + 1$

When $n \geq 5$,

By definition:

$c_1 g(n) \leq f(n) \leq c_2 g(n) \;\; for\ all\ n \geq n_0$

We get:

$5n^2 \leq 5n^2 + 3n + 1 \leq 6n^2 [6n^2 > 5n^2 and\ 4n^2 < 5n^2$

$the\ degree\ of\ polynomial]$

$Here, c_1 = 5 , c_2 = 6 , g(n) = n^2, n_0 = 5$

Hence , $5n^2 + 3n + 1 = \theta(n^2)$

EXAMPLE 6

Let us consider that $f(n) = n^4 + 3n^3 + 5n + 1\ and$

$g(n) = n^4 + 1$, Prove that f(n) of an algorithm is in

$\Theta(n^4)$.

Solution:

By definition:

$$c_1 g(n) \leq f(n) \leq c_2 g(n) \;\; for\, all\, n \geq n_0$$

$$n^4 \leq n^4 + 3n^3 + 5n + 1 \leq 2n^4$$

Hence

$$c_1 = 1 \; and \; c_2 = 2$$

It is also true for

$$1 \times (n^4 + 1) \leq n^4 + 3n^3 + 5n + 1 \leq 2 \times (n^4 + 1)$$

It can be observed that this condition holds good for

$c_1 = 1 \;\; and \; c_1 = 2 \,. Therefore \,, f \; is \; in \; \Theta(n)$.

In other words, the algorithm complexity is in $\Theta(n^4)$.

EXAMPLE 7

Find theta notation for $f(n) = 3n^2 + 2n + 5$

Solution

As per definition of $'\Theta'$ notation, the function $f(n)$ such that

$$\Rightarrow c_1 \times g(n) \leq 3n^2 + 2n + 5 \leq c_2 \times g(n)$$

When $n \geq 3$

$\Rightarrow 3n^2 \leq 3n^2 + 2n + 5 \leq 4n^2$

Hence it puts:

$c_1 \times g(n) \leq 3n^2 + 2n + 5, for\ all\ n \geq n_0$

$Again,$

$3n^2 + 2n + 5 \leq c_2 \times g(n), for\ all\ n \geq n_0$

$Therefore, 3n^2 + 2n + 5 = \Theta(n^2)$

EXAMPLE 8

Prove $f(n) = n^4 + 3n^3 = \Theta(n^4)$

Solution

As per definition of $'\Theta'$ notation, the function $f(n)$ such that

$\Rightarrow c_1 \times g(n) \leq n^4 + 3n^3 \leq c_2 \times g(n)$

When $n \geq 1$

$\Rightarrow n^4 \leq n^4 + 3n^3 \leq 2n^4\ , for\ all\ n \geq n_0$

$Hence\ c_1 = 1\ , c_2 = 2\ , n_0 = 1$ and $g(n) = n^4$

$And\ f = \Theta(g) \Rightarrow f(n) = \Theta\big(g(n)\big) \Rightarrow \Theta(\mathrm{n}^4)$

EXAMPLE 9

Prove $log(n!) = \Theta(nlog(n))$

Solution

$$log(n!) = log(1) + log(2) + log(3) + \cdots + log(n)$$

Upper Bound

$$log(1) + log(2) + log(3) + \cdots + log(n) \\ \leq log(n) + log(n) + log(n) + \cdots + log(n)$$

[As $log(1) \leq log(n), log(2) \leq log(n), \ldots\ldots, log(n) \leq log(n)$]

$$log(1) + log(2) + log(3) + \cdots + log(n) \leq nlog(n)$$

Lower Bound

$$(n!)^2 \geq \prod_{k=1}^{n} n_k$$

$$\left[\prod_{k=1}^{n} a_k = a_0 \times a_1 \times \ldots \times a_n\right]$$

$$\left[\begin{array}{l} \prod_{k=1}^{n} a_k \text{ , if } a = 6 \text{ , } \quad n = 2 \\ a_1 = 6 \times a_2 = 6 = 36 => 6\text{^}2 \end{array}\right]$$

$$(n!)^2 \geq n^n \left(\prod_{k=1}^{n} n_k = n^n\right)$$

$\textit{Putting log in LHS and RHS we get}$:

$$log(n!)^2 \geq logn^n$$

$$2log(n!) \geq nlogn$$

$$log(n!) \geq \frac{1}{2}nlogn$$

Which stands like :

$$\frac{1}{2}(nlogn) \leq log(n!) \leq nlogn$$

Hence $c_1 = \frac{1}{2}$, $c_2 = 1$, and $g(n) = nlogn$

$And\ f = \Theta(g) \Rightarrow f(n) = \Theta\big(g(n)\big) \Rightarrow \Theta(nlogn)$

EXAMPLE 10

Suppose that an algorithm that an algorithm takes eight seconds to run on an input size n = 12 . Estimate the instances that can be processed in 56 seconds . Assume that the algorithm complexity is $\boldsymbol{\Theta}(\boldsymbol{n})$.

Solution

Assume that the time complexity is $\boldsymbol{\Theta}(\boldsymbol{n})$, then $\boldsymbol{cn} \approx \mathbf{8}$ seconds . Here the instance $\boldsymbol{n}$ is given as $\mathbf{12}$.

Therefore $\mathbf{12}\ \boldsymbol{c} = \mathbf{8}$; $\boldsymbol{hence}$, $\boldsymbol{c} = \frac{\mathbf{8}}{\mathbf{12}} = \frac{\mathbf{2}}{\mathbf{3}}$.

The problem is to determine the value of $\boldsymbol{n}$ that can be processed in 56 seconds . This implies that $\boldsymbol{c} \times \boldsymbol{n} = \mathbf{56}$.

The value of $\boldsymbol{c}$ has already been determined as $\frac{\mathbf{2}}{\mathbf{3}}$. Therefore , $\frac{\mathbf{2}}{\mathbf{3}} \times \boldsymbol{n} = \mathbf{56}$. This implies that $\boldsymbol{n} = \mathbf{56} \times \frac{\mathbf{3}}{\mathbf{2}} = \mathbf{84}$.

Hence, the maximum input that is possible is 84.