MODULE 1

Faculty : Dr. Reshmi R

January 8, 2025

ASYMPTOTIC NOTATIONS

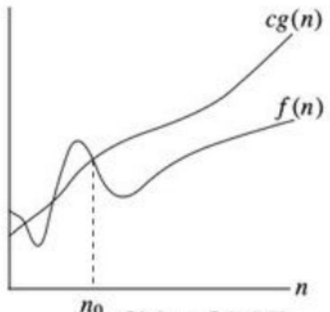
Terminology that enable to make meaningful statements about the time and space complexities of a program.

1. Big 'oh'- $\{O\}$: Upper bound of the algorithm/**Worst case**
2. Omega - $\{\Omega\}$: Lower bound of the algorithm/**Best case**
3. Theta – $\{\theta\}$: Average bound of the algorithm/**Average case**

ASYMPTOTIC NOTATIONS

Big Oh- $\{O\}$

$f(n) = O(g(n))$ or $(f(n) \in O(g(n)))$ iff there exist positive constants $c > 0$ and n_0 such that $f(n) \leq c.g(n)$ for all n, $n \geq n_0$.



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

Big Oh- $\{O\}$

Examples :

1. $3n+2 = O(n)$ as $3n+2 \leq 4n$ for all $n \geq 2$
2. $3n+3 = O(n)$ as $3n+3 \leq 4n$ for all $n \geq 3$
3. $100n+6 = O(n)$ as $100n+6 \leq 101n$ for all $n \geq 10$
4. $10n^2+4n+2 = O(n^2)$ as $10n^2+4n+2 \leq 11n^2$ for $n \geq 5$
5. $f(n)$ is a polynomial of degree k then $f(n) = O(n^k)$

Big Oh- $\{O\}$

Eg:

Prove that $3n + 2 = O(n)$

Let $f(n) = 3n + 2$ and $g(n) = n$

Definition of $Big'O'$ says, $f(n) \leq c * g(n)$

$\therefore 3n + 2 = O(n)$

So, we can write $3n + 2 \leq 4n$ for all $n \geq 2$

Let $c = 4$

$n = 0,\ 3 * 0 + 2 \nleq 4 * 0$

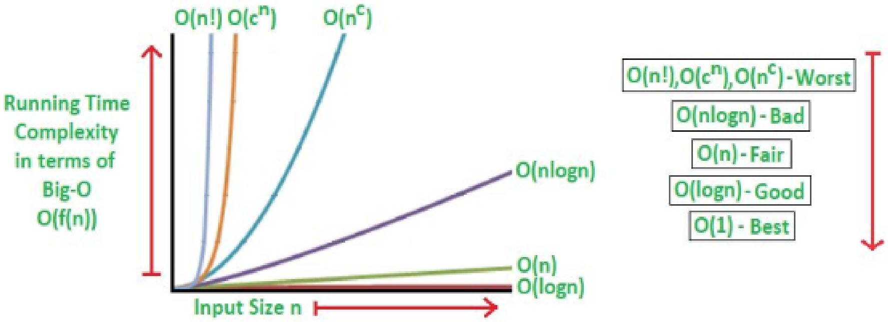
$n = 1,\ 3 * 1 + 2 \nleq 4 * 1$

$n = 2,\ 3 * 2 + 2 \leq 4 * 2$

Big Oh- $\{O\}$

1. $O(1)$ – constant computing time
2. $O(n)$ – linear time complexity
3. $O(n^2)$ – quadratic time complexity
4. $O(n^3)$ – cubic time complexity
5. $O(2^n)$ – exponential time complexity
6. $O(logn)$ – logarithmic time complexity

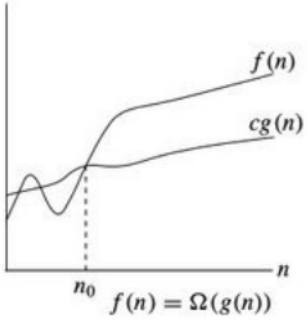
Big Oh- $\{O\}$



O(n!) O(c^n)
O(n^c)
Running Time Complexity in terms of Big-O O(f(n))
O(nlogn)
O(n)
O(logn)
Input Size n
O(n!),O(c^n),O(n^c) - Worst
O(nlogn) - Bad
O(n) - Fair
O(logn) - Good
O(1) - Best

Omega- $\{\Omega\}$

$f(n) = \Omega(g(n))$ or $(f(n) \in \Omega(g(n)))$ iff there exist positive constants $c > 0$ and n_0 such that $f(n) \geq c.g(n)$ for all n, $n \geq n_0$.



Omega- $\{\Omega\}$

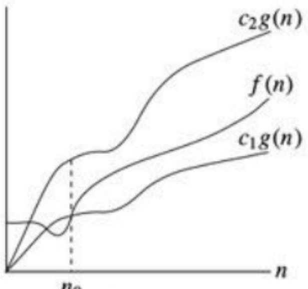
Examples :

1. $3n + 2 = \Omega(n)$ as $3n + 2 \geq 3n$ for all $n \geq 1$
2. $3n + 3 = \Omega(n)$ as $3n + 3 \geq 3n$ for all $n \geq 1$
3. $100n + 6 = \Omega(n)$ as $100n + 6 \geq 100n$ for all $n \geq 1$
4. $10n^2 + 4n + 2 = \Omega(n^2)$ as $10n^2 + 4n + 2 \geq 10n^2$ for $n \geq 1$

ASYMPTOTIC NOTATIONS

Theta- $\{\theta\}$

$f(n) = \theta(g(n))$ iff there exist positive constants c_1, c_2 and n_0 such that $c_1.g(n) \leq f(n) \leq c_2.g(n)$ for all n, $n \geq n_0$.



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

Theta- $\{\theta\}$

Example :

1. $3n + 2 = \theta(n)$ as $3n + 2 \geq 3n$ for all $n \geq 2$ and $3n + 2 \leq 4n$ for all $n \geq 2$.

Its more precise than O & Ω.

ASYMPTOTIC NOTATIONS

There are some other notations present except the Big-Oh, Big-Omega and Big-Theta notations. They are

- little oh (o)
  Little o notation is used to describe an upper bound that cannot be tight. In other words, loose upper bound of $f(n)$.
- little omega (ω)
  Little omega (ω) notation is used to describe a loose lower bound of $f(n)$.

ASYMPTOTIC NOTATIONS - Little oh(o)

Let $f(n)$ and $g(n)$ are the functions that map positive real numbers. We can say that the function $f(n)$ is $o(g(n))$ or $(f(n) \in o(g(n)))$, if for any real positive constant $c > 0$, there exists an integer constant $n_0 \geq 1$ such that $f(n) < c * g(n)$ for every integer $n \geq n_0$.

Mathematical Relation of Little o notation

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

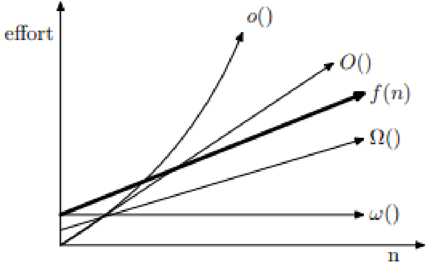
ASYMPTOTIC NOTATIONS - Little omega(ω)

Let $f(n)$ and $g(n)$ are the functions that map positive real numbers. We can say that the function $f(n)$ is $\omega(g(n))$ or $(f(n) \in \omega(g(n)))$, if for any real positive constant $c > 0$, there exists an integer constant $n_0 \geq 1$ such that $f(n) > c * g(n)$ for every integer $n \geq n_0$.

Mathematical Relation of Little omega notation

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

ASYMPTOTIC NOTATIONS



$f(n) \in \omega(g(n))$ if and only if $f(n) \in o(g(n))$.

Definition	$\boxed{?}\ c > 0$	$\boxed{?}\ n_0 \geq 1$	$f(n)\ \boxed{?}\ c \cdot g(n)$
$O()$	$\exists$	$\exists$	$\leq$
$o()$	$\forall$	$\exists$	$<$
$\Omega()$	$\exists$	$\exists$	$\geq$
$\omega()$	$\forall$	$\exists$	$>$

Figure: Summary table

ASYMPTOTIC NOTATIONS-Example

1. Let $f(n) = 7n + 8$ and $g(n) = n$. Is $f(n) \in O(g(n))$?

   For $7n + 8 \in O(n)$, we have to find c and n_0 such that $7n + 8 \leq c \cdot n$, $\forall n \geq n_0$. By inspection, it's clear that c must be larger than 7. Let $c = 8$.

   Now we need a suitable n_0. In this case, $f(8) = 8 \cdot g(8)$. Because the definition of $O()$ requires that $f(n) \leq c \cdot g(n)$, we can select $n_0 = 8$, or any integer above 8 – they will all work.

   We have identified values for the constants c and n_0 such that $7n + 8$ is $\leq c \cdot n$ for every $n \geq n_0$, so we can say that $7n + 8$ is $O(n)$.

   (But how do we know that this will work for every n above 7? We can prove by induction that $7n+8 \leq 8n$, $\forall n \geq 8$. Be sure that you can write such proofs if asked!)

ASYMPTOTIC NOTATIONS-Example

2. Let $f(n) = 7n + 8$ and $g(n) = n$. Is $f(n) \in o(g(n))$?

   In order for that to be true, for any c, we have to be able to find an n_0 that makes $f(n) < c \cdot g(n)$ asymptotically true.

   However, this doesn't seem likely to be true. Both $7n + 8$ and n are linear, and $o()$ defines loose upper-bounds. To show that it's not true, all we need is a counter–example.

   Because any $c > 0$ must work for the claim to be true, let's try to find a c that won't work. Let $c = 100$. Can we find a positive n_0 such that $7n + 8 < 100n$? Sure; let $n_0 = 10$. Try again!

   Let's try $c = \frac{1}{100}$. Can we find a positive n_0 such that $7n + 8 < \frac{n}{100}$? No; only negative values will work. Therefore, $7n + 8 \notin o(n)$, meaning $g(n) = n$ is not a loose upper-bound on $7n + 8$.

ASYMPTOTIC NOTATIONS-Example

3. Is $7n + 8 \in o(n^2)$?

   Again, to claim this we need to be able to argue that for any c, we can find an n_0 that makes $7n+8 < c \cdot n^2$. Let's try examples again to make our point, keeping in mind that we need to show that we can find an n_0 for any c.

   If $c = 100$, the inequality is clearly true. If $c = \frac{1}{100}$, we'll have to use a little more imagination, but we'll be able to find an n_0. (Try $n_0 = 1000$.) From these examples, the conjecture *appears* to be correct.

   To prove this, we need calculus. For $g(n)$ to be a loose upper-bound on $f(n)$, it must be the case that $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$. Here, $\lim_{n\to\infty} \frac{7n+8}{n^2} = \lim_{n\to\infty} \frac{7}{2n} = 0$ (by l'Hôpital). Thus, $7n + 8 \in o(n^2)$.

ASYMPTOTIC NOTATIONS- Examples

```
sum = 0   ------------------------1
for (i=0;i<n;i++) ----------------n
        sum ++ ---------1
```

O(n)

$\sum_{i=0}^{n-1} 1 = n = O(n)$

```
sum = 0   ------------------------1
for (i=0;i<n;i++) ----------------n
for (j=0;j<n;j++) ----------------n
        sum ++ ---------1
```

$O(n^2)$

$\sum_{i=0}^{n-1} 1 \sum_{i=0}^{n-1} 1 = \sum_{i=0}^{n-1} 1(n) = n \sum_{i=0}^{n-1} 1 = n(n) = n^2 = O(n^2)$

ASYMPTOTIC NOTATIONS- Examples

```
sum = 0   ------------------------1
for (i=0;i<n;i++) ----------------n
for (j=0;j<n;j++) ----------------n
for (k=0;k<n;k++) ----------------n
      sum ++ ---------1
```

$O(n^3)$

$$\sum_{i=0}^{n-1} 1 \sum_{i=0}^{n-1} 1 \sum_{i=0}^{n-1} 1 = \sum_{i=0}^{n-1} 1 \sum_{i=0}^{n-1} 1 (n) = n \sum_{i=0}^{n-1} 1 \sum_{i=0}^{n-1} 1 = n^2 \sum_{i=0}^{n-1} 1 = n^3 = O(n^3)$$

ASYMPTOTIC NOTATIONS- Examples

```
sum = 0   ------------------------1
for (i=0;i<n;i++) ----------------n
for (j=0;j<n*n;j++) ----------------n^2
      sum ++ ---------1
```

$O(n^3)$

$$\Sigma_{i=0}^{n-1} 1 \Sigma_{i=0}^{n^2-1} 1 = \Sigma_{i=0}^{n-1} 1(n^2) = n^2 \Sigma_{i=0}^{n-1} 1 = n^3 = O(n^3)$$

ASYMPTOTIC NOTATIONS- Examples

$O(n^3)$

```
void function(int n)
{
   for(int i = 0; i < n*n; i++)  -----------n^2
      for (int j = 0; j < n; j++) ------------n
         "something taking O(1) time"-----1
}
```

$O(n^3)$

```
void function(int n)
{
   for(int i = 0; i < n*n; i++)  -----------n^2
      for (int j = 0; j < n/2; j++) ------------n
         "something taking O(1) time"-----1
}
```

ASYMPTOTIC NOTATIONS- Examples

```
void function(int n)                                O(n logn)
{
   for(int i = 0; i < n; i++)  ------------n
      for (int j = 1; j < n; j*=2) -------------logn
         "something taking O(1) time"-----1
}
```

A loop running through i = 1, 2, 4, ..., n runs O(log n) times!