# Advanced Algorithms

**Algorithm**: Finite set of clear and unambiguous instructions that accomplishes a particular task.

**Properties**:
- Inputs (zero/more no.)
- Outputs (Atleast one)
- Definiteness (clear & unambiguous)
- Finiteness (after a finite no. of steps)
- Effectiveness (basic instructions)

**Distinct areas of studying DAA**:

- How to design an algorithm
- How to analyse an algorithm (Time Complexity / Space ")
- How to validate an algorithm (Proof of correctness)

Design of fast algorithm

G.C.D finding problem

(Greatest Common Divisor)

**Problem statement**: Given two +ve integers m & n, find their G.C.D.

## Algorithm 1 (Factoring)

I/P: Two +ve integers m & n

O/P: Largest +ve integer that divides both

1. Factorize m: Find primes $m_1, m_2, \ldots, m_k$ s.t
   $m = m_1 \times m_2 \times \ldots \times m_k$
2. Factorize n:
   $n = n_1 \times n_2 \times \ldots \times n_j$
3. Identify common factors
4. Multiply & return result.

## Algorithm 2 (Euclid's)

```
Euclid (m, n)
{// m, n are two +ve integers
 while m does not divide n
     r = n mod m
     n = m
     m = r
 end while
 return m
}
```

---

$m = 36, n = 48$

$m = 2 \times 2 \times 3 \times 3$ (36)

$n = 2 \times 2 \times 2 \times 2 \times 3$ (48)

G.C.D $= 2 \times 2 \times 3 = 12$

Divisions: $4 + 5 = 9$

---

$m = 36, n = 48$

→ 36 divides 48? → No

$r = 48 \bmod 36 = 12$

$n = 36, m = 12$

→ 12 divides 36? → Yes

G.C.D = 12

Divisions: 2

---

$m = 434, n = 966$

$434 = 2 \times 7 \times 31$

$966 = 2 \times 3 \times 7 \times 23$

G.C.D $= 2 \times 7 = 14$

Divisions: Many

---

$m = 434, n = 966$

→ 434 divides 966? → No

$r = 966 \bmod 434 = 98$

$n = 434, m = 98$

→ 98 divides 434? → No

$r = 434 \bmod 98 = 42$

$n = 98, m = 42$

→ 42 divides 98? → No

$r = 98 \bmod 42 = 14$

$n = 42, m = 14$

→ 14 divides 42? → Yes

G.C.D = 14

Divisions: 4

Validation of algorithm ← Proof of correctness of Euclid's Algo^m for finding G.C.D

→ We have to prove that for all valid input instances the algorithm works correctly.

Facts:
- If m divides n, then GCD(m, n) = m
- If m does not divide n, then GCD(m, n) = GCD(n mod m, m)

As the loop executes, m and n might change but their GCD does not change.

In each iteration of while loop, m is decreasing. It will have to decrease by atleast 1 in each iteration. m can be zero after several decrements.

So, ultimately loop will exit and if the loop exits the correct GCD value will be returned.

---

Asymptotic notation: Formal way of classifying functions and classify them.

1. Theta notation ($\Theta$)
2. Big Oh " ($O$)
3. Big Omega " ($\Omega$)
4. Little Oh notation ($o$)
5. Little Omega " ($\omega$)

"Asymptotic tight bound"

1. $\Theta$ notation: Let, $f$ and $g$ be two non-negative functions. $\Theta(g(n)) = \{$Set of functions $f(n)$ such that there exists +ve constants $c_1, c_2$ & $n_0$ so that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$, for $n \geq n_0$ holds$\}$

"Asymptotic upper bound"

2. $O$ notation: $c, n_0 \Rightarrow$ +ve constants.

$O(g(n))$ $\quad 0 \leq f(n) \leq c g(n)$, for $n \geq n_0$

"Asymptotic lower bound"

3. $\Omega$ notation: $c, n_0 \Rightarrow$ +ve constants

$\Omega(g(n))$ $\quad 0 \leq c g(n) \leq f(n)$, for $n \geq n_0$

Problem-1: given $\begin{cases} f(n) = 10n^2 + 4n + 2 \\ g(n) = n^2 \end{cases}$

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

$10n^2 + 4n + 2 \leq 11n^2 \quad \forall n \geq 5$ ($n_0$)

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

$10 \cdot 5^2 + 4 \cdot 5 + 2 \leq 11 \cdot 5^2$ ✓

$f(n) = O(g(n))$ where $c = 11, n_0 = 5$ (Proved)

Problem-2: Same as prob-1 but check using $\Omega$ notation

$cg(n) \leq f(n) \Rightarrow f(n) \geq c\,g(n)$

$10n^2 + 4n + 2 \geq 1 \cdot n^2, \quad \forall n \geq 1$ ($n_0$)

$10 \cdot 1^2 + 4 \cdot 1 + 2 \geq 1 \cdot 1^2$ ($c$)

$f(n) = \Omega(g(n))$ where $c = 1$ and $n_0 = 1$ (Proved)

Prob-3: Same as problem-1, but using $\Theta$ notation.

$f(n)$ $g(n)$

✓ $10n^2 + 4n + 2 \leq 11n^2$, $\forall n \geq 5$

$c_2$ $n_0$

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

✓ $10n^2 + 4n + 2 \geq 1 \cdot n^2$, $\forall n \geq 5$

$c_1$ $n_0$

$f(n) = \Theta(g(n))$ where $c_1 = 1$, $c_2 = 11$, $n_0 = 5$

(Proved)

Exercise: 1. $f(n) = 6 \cdot 2^n + n^2$

$g(n) = 2^n$

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

2.
```
for(i=0; i<n; i++)
{
    for(j=1; j<n; j=j*2)
    {
        a = b+c;
    }
}
```

Find out the time complexity in O notation.

3. Write an algorithm to multiply two matrices and show its time and space complexity.

1 o-notation (Little oh): $f, g \rightarrow$ two non-negative functions.

$o(g(n)) = \{$ set of functions $f(n)$ s.t. for any +ve constant $c$ and $n_0$, $0 \leq f(n) < cg(n)$ holds, for $n \geq n_0\}$

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

Ex-1: $f(n) = n$, $g(n) = n^2$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{n}{n^2} = \lim_{n \to \infty} \frac{1}{n} = \frac{1}{\infty} = 0 \text{ (zero)}$$

$f(n) = o(g(n))$ (Proved)

Ex-2: $f(n) = \log n$, $g(n) = n$

$$\lim_{n \to \infty} \frac{\log n}{n} = \frac{\log \infty}{\infty}$$

Applying L'Hospital's rule,

$$\lim_{n \to \infty} \frac{f'(n)}{g'(n)} = \lim_{n \to \infty} \frac{1/n}{1} = \lim_{n \to \infty} \frac{1}{n} = \frac{1}{\infty} = 0 \text{ (zero)}$$

$f(n) = o(g(n))$ (Proved)

ω-notation (Little omega): $f, g \Rightarrow$ two non-negative fun$^n$.

$\omega(g(n)) = \{$ set of functions $f(n)$ s.t. for +ve constants $c$ and $n_0$, $0 \leq cg(n) < f(n)$, holds $\forall\, n \geq n_0\}$

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

Ex-1: $f(n) = 3^n$, $g(n) = 2^n$

$$\lim_{n\to\infty} \frac{3^n}{2^n} = \lim_{n\to\infty} \left(\frac{3}{2}\right)^n = \infty$$

$\therefore f(n) = \omega(g(n))$ (Proved)

Ex-2: $f(n) = n^2$, $g(n) = \log n$

$$\lim_{n\to\infty} \frac{n^2}{\log n} = \frac{\infty}{\infty}$$

Applying L Hospital's rule,

$$\lim_{n\to\infty} \frac{f'(n)}{g'(n)} = \lim_{n\to\infty} \frac{2n}{1/n} = \lim_{n\to\infty} 2n^2 = \infty$$

$\therefore f(n) = \omega(g(n))$ (Proved)

Exercise: • check whether $2n^2 = o(n^2)$
• Check whether $\frac{n^2}{2} = \omega(n^2)$

Properties of asymptotic notation:

- Transitivity: $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$ then $f(n) = \Theta(h(n))$

  Same is true for $O$, $\Omega$, $o$, $\omega$ notations
- Reflexivity: $f(n) = \Theta(f(n))$, $f(n) = O(f(n))$, $f(n) = \Omega(f(n))$
- Symmetry: $f(n) = \Theta(g(n))$ iff $g(n) = \Theta(f(n))$

## Proof of symmetry property:

**Necessary part:** $f(n) = \Theta(g(n))$ then $g(n) = \Theta(f(n))$

Basic def$^n$ of $\Theta$ notation,

$c_1 g(n) \leq f(n) \leq c_2 g(n)$ , $c_1, c_2 \rightarrow$ +ve const.

$\hookrightarrow \Rightarrow g(n) \leq \frac{1}{c_1} \cdot f(n)$ and $g(n) \geq \frac{1}{c_2} \cdot f(n)$

$\Rightarrow \frac{1}{c_2} \cdot f(n) \leq g(n) \leq \frac{1}{c_1} \cdot f(n)$

$g(n) = \Theta(f(n))$ , [$\frac{1}{c_1}$ and $\frac{1}{c_2}$ are well defined]

**Sufficiency part:** $g(n) = \Theta(f(n))$ then $f(n) = \Theta(g(n))$

Basic def$^n$ of $\Theta$, $c_1 f(n) \leq g(n) \leq c_2 f(n)$

$\Rightarrow f(n) \leq \frac{1}{c_1} \cdot g(n)$ and $f(n) \geq \frac{1}{c_2} \cdot g(n)$

$\Rightarrow \frac{1}{c_2} \cdot g(n) \leq f(n) \leq \frac{1}{c_1} \cdot g(n)$

So, $f(n) = \Theta(g(n))$ (Proved)

So, symmetry property is proved.

## Proof of transitivity property:

$f(n) = O(g(n))$ and $g(n) = O(h(n)) \Rightarrow f(n) = O(h(n))$

**Proof:** $f(n) \leq c_1 g(n)$ $\quad$ $g(n) \leq c_2 h(n)$

$f(n) \leq c_1 c_2 h(n)$

$f(n) \leq c\, h(n)$ [$\because c = c_1 \cdot c_2$]

(Proved)

## Transpose symmetry:

- $f(n) = O(g(n))$ iff $g(n) = \omega(f(n))$
- ✓ $f(n) = O(g(n))$ iff $g(n) = \Omega(f(n))$

**Proof**: **Necessary part**:

$f(n) = O(g(n))$ we have to prove $g(n) = \Omega(f(n))$

$\hookrightarrow f(n) \leq c\,g(n)$

$g(n) \geq \frac{1}{c} \cdot f(n)$

So, $g(n) = \Omega(f(n))$ (Proved)

**Sufficiency part**: $g(n) = \Omega(f(n))$ we have to prove $f(n) = O(g(n))$

$\hookrightarrow g(n) \geq c\,f(n)$

$f(n) \leq \frac{1}{c} \cdot g(n)$

So, $f(n) = O(g(n))$ (Proved)

So, transpose symmetry property is proved.