**Chapter 1: ASYMPTOTIC NOTATIONS**

**Topic – 1: Introduction**

* When runtime equations are represented on a **graph**, they provide the **best insight** into performance of codes.
* Formally equations are defined with theta (**θ**) of the **most dominant** term.
* And graphically, equation with **less dominant** term **outperforms** the one with **more dominant** term.
* For example, **θ(n2)** performs better than **θ(n3)**.
* Goal of ***asymptotic notations*** is to compare performance of the code, **independent** of the machine being used.

**Topic – 2: Big Theta (θ)**

**Equation**

**Θ(f(n)) = {g(n) | c1f(n) ≤ g(n) ≤ c2f(n) | n ≥ n0}**

* **c1** and **c2** are two **+ve constants**.
* We also call it as ***bounded between***.

**Example**

**T(n) = 10n2 + 5 = Θ(n2)**

**Θ(n2) = {g(n) | c1n2 ≤ g(n) ≤ c2n2}**

**Topic – 3: Big O**

**Equation**

**O(f(n)) = {g(n) | g(n) ≤ cf(n) | n ≥ n0}**

* **c** is a **+ve** constant.
* We also call it ***bounded above***.

**Example**

**T(n) = 10n2 - 5n + 2 = O(n2)**

**O(n2) = {g(n) | g(n) ≤ cn2}**

**Topic – 4: Big Omega (Ω)**

**Equation**

**Ω(f(n)) = {g(n) | cf(n) ≤ g(n) | n ≥ n0}**

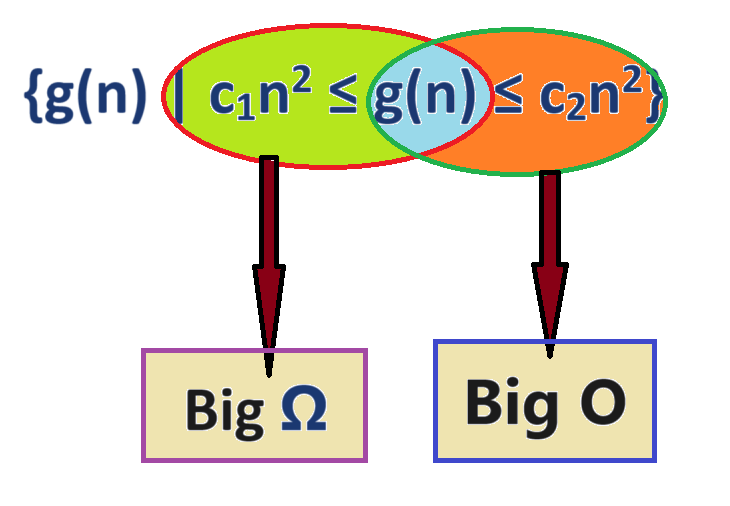
* **c** is a **+ve** constant.
* We also call it as ***bounded below***.

**Example**

**T(n) = 10n2 + 5 = Ω(n2)**

**Ω(n2) = {g(n) | cn2 ≤ g(n)}**

**Facts Revealed**



* See the **big theta**’s equation above.
* So, when an equation is **θ(f(n))** then it is also **O(f(n))** and **Ω(f(n))**.
* We usually go for **upper bound**, because that tells us the **maximum time** the program will take to execute, which is more useful.