**ASYMPTOTIC NOTATIONS**

**Introduction to Big O**

* Simplified analysis of an algorithm’s efficiency.
* Machine independent.
* Used for calculating time & space complexity.

**Big O General Rules**

* Ignores constants **[5n -> O(n)]**.
* Certain terms dominate others:
  + **O(1) < O(log n) < O(n) < O(n log n) < O(n2) < O (2n) < O(n!)**

**Time Complexity Terms**

* **O(1) –** Constant time
* **O(log n) –** Logarithmic time
* **O(n) –** Linear time
* **O(n log n) –** Quasilinear time
* **O(n2) –** Quadratic time
* **O(2n) –** ?
* **O(n!) –** Factorial time

**Big O Graph**

* Also called **upper bound**.
* Equation: **0 ≤ f(n) ≤ cg(n)**

**Ω Notation**

* Also called **big omega** and **lower bound**.
* Equation: **0 ≤ cg(n) ≤ f(n)**

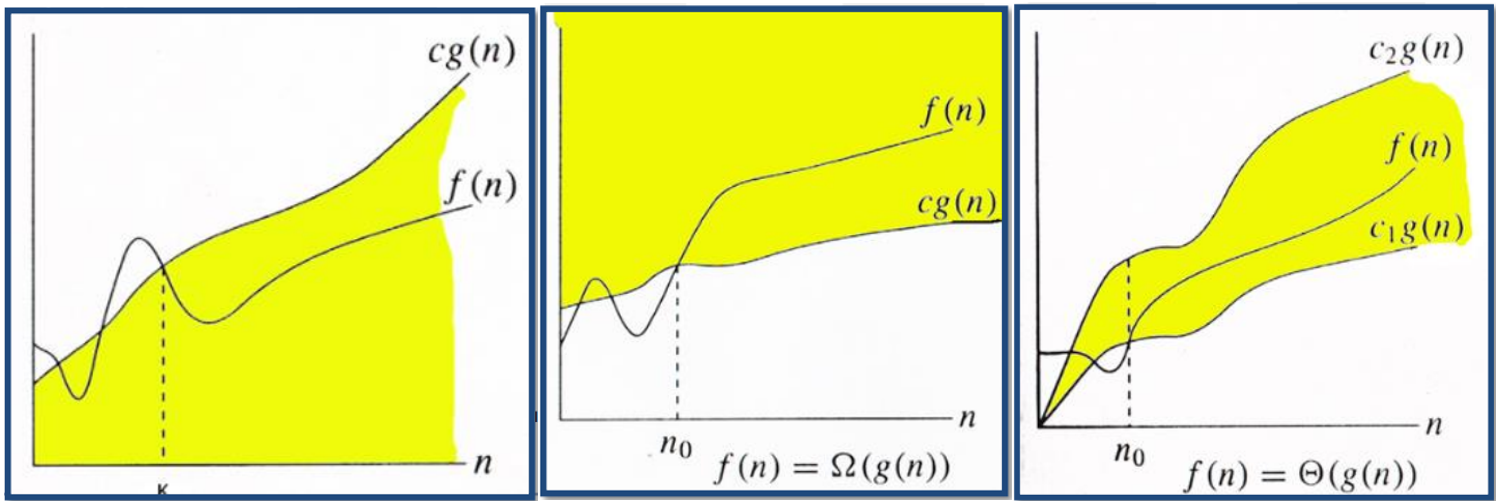
**θ Notation**

* Also called **theta notation**.
* Curve **f(n)** is sandwiched between curve **c1g(n)** and **c2g(n)**.
* Equation: **0 ≤ c1g(n) ≤ f(n) ≤ c2g(n)**

**General Information**

* We write **f(n) = notation(g(n))** when equation is satisfied with a notation.
* Above, **notation** can be **O** or **Ω** or **θ** etc.
* **f(n)** always intersects with **“c”** line(s) at some point.
* **Other notations include:**
  + Small O: **o**
  + Small omega: **ω**

**Graphs of Various Notations**



* First one: **Big O**
* Second one: **Big omega**
* Third one: **Theta**