# Analysis of Algorithm

## Amypo Technologies Pvt Ltd

# Agenda

- Algorithmic Efficiency
- Analysis Framework
- Asymptotic Notations and their properties
- Time Complexity Analysis

# Analysis Framework

- The efficiency of an algorithm can be decided by measuring the performance of an algorithm. We can measure the performance of an algorithm by computing two factors.

    1. Amount of time required by an algorithm to execute.

    2. Amount of storage required by an algorithm.

- This is popularly known as time complexity and space complexity of an algorithm.

# Asymptotic Analysis of Complexity Bounds

- To choose the best algorithm, we need to check efficiency of each algorithm. The efficiency can be measured by computing time complexity of each algorithm. Asymptotic notation is a shorthand way to represent the time complexity.

- Using asymptotic notations we can give time complexity as "fastest possible", "slowest possible" or "average time".

- Various notations such as Ω, Θ and O used are called asymptotic notions.

# Big oh Notation

- The Big oh notation is denoted by 'O'. It is a method of representing the upper bound of algorithm's running time. Using big oh notation we can give longest amount of time taken by the algorithm to complete.

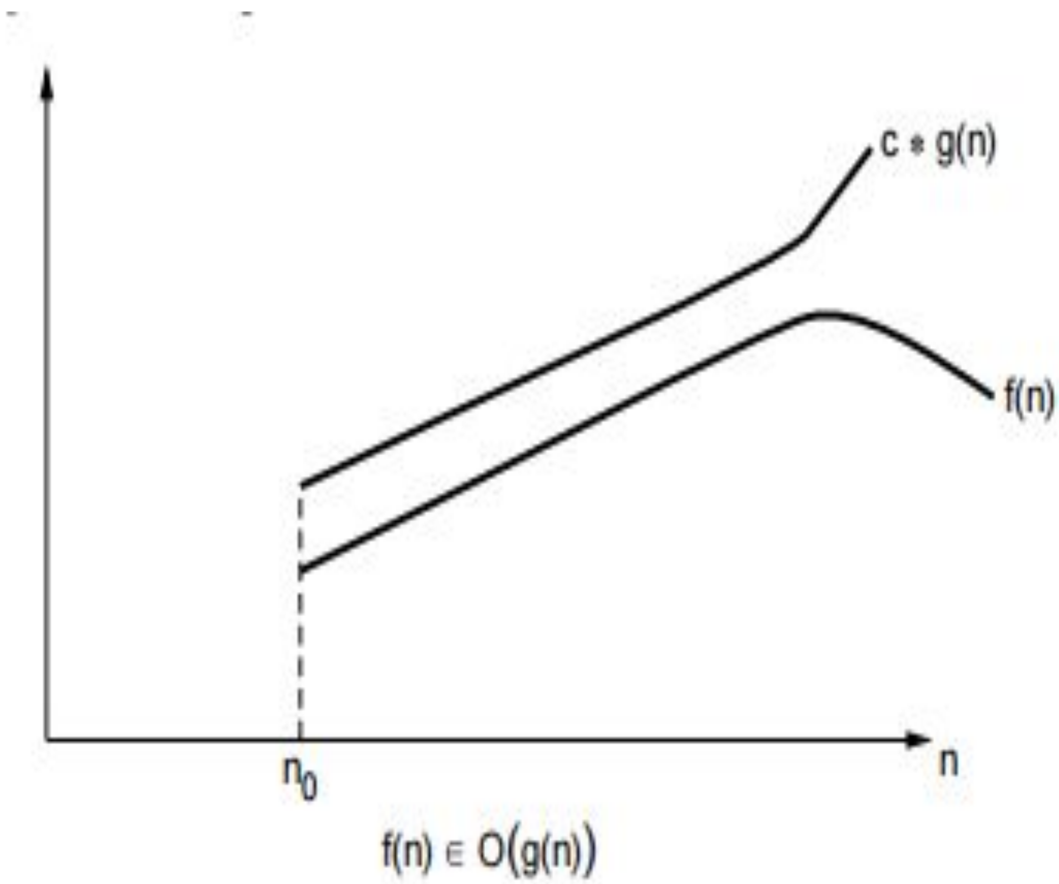
## Definition

- Let f(n) and g(n) be two non-negative functions.
- Let no and constant c are two integers such that n_{0} denotes some value of input and n>n_{0} Similarly c is some constant such that c>0 We can write

  **f(n)<= c * g(n)**

- then f(n) is big oh of g(n). It is also denoted as f(n)\in O(g(n)) In other words f(n) is less than g(n) if g(n) is multiple of some constant c.

# Big oh Notation



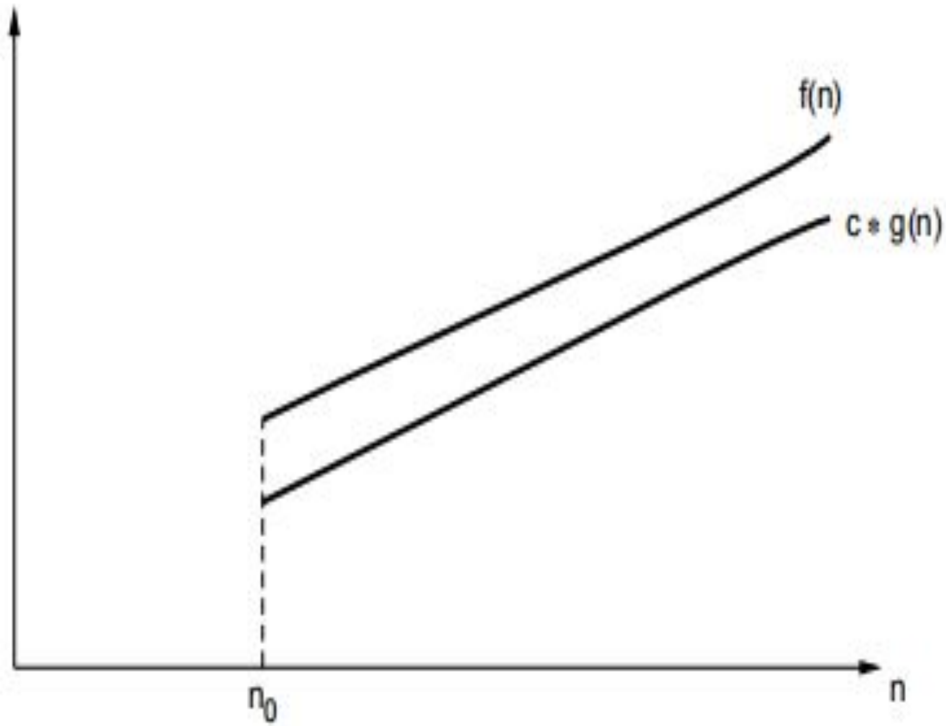
## Omega Notation

- Omega notation is denoted by 'Ω'. This notation is used to represent the lower bound of algorithm's running time. Using omega notation we can denote shortest amount of time taken by algorithm.

## Definition

- A function f(n) is said to be in 2 (g(n)) if f(n) is bounded below by some positive constant multiple of g(n) such that
- f(n)>= c*g(n)    For all n\ge n_{0}
- It is denoted as f(n)\in\Omega(g(n)) Following graph illustrates the curve for Ω notation

# Omega Notation




$f(n) \in \Omega(g(n))$

# Theta Notation

The theta notation is denoted by e. By this method the running time is between upper bound and lower bound.

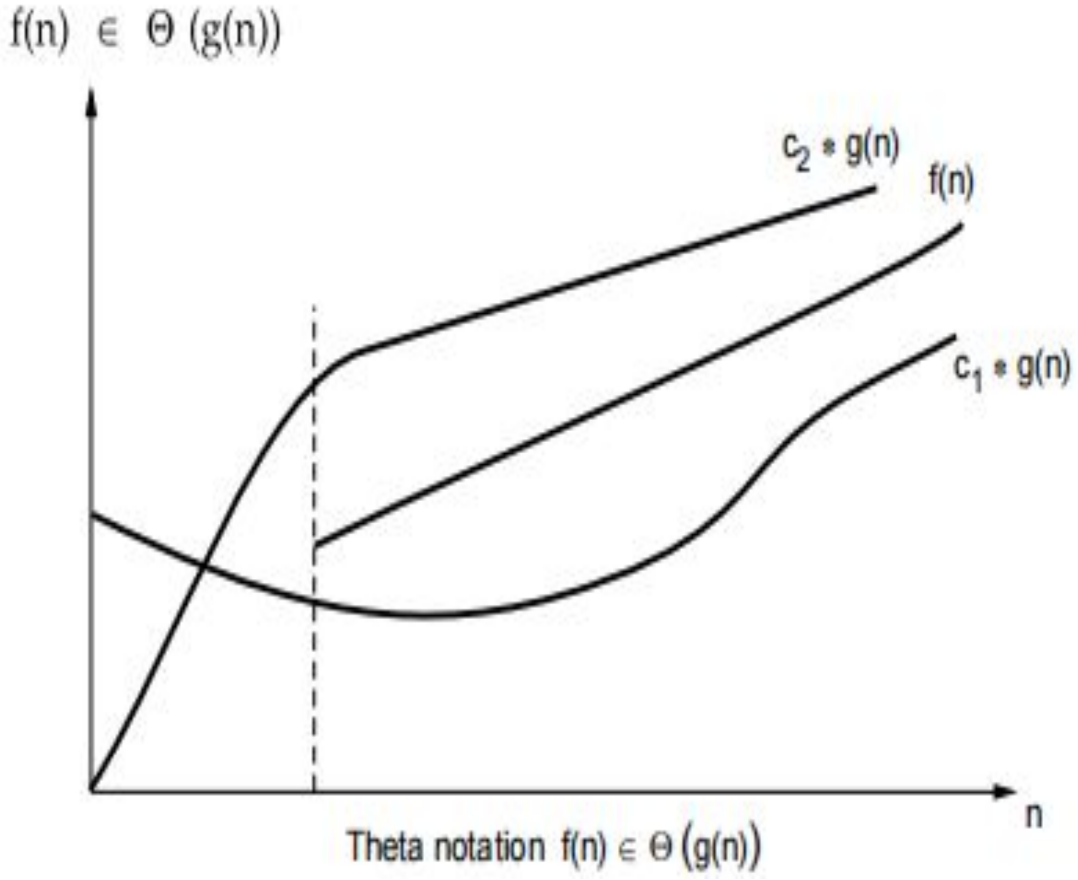
## Definition

- Let f(n) and g(n) be two non negative functions. There are two positive constants namely $c_{1}$ and $c_{2}$ such that

$$c_{1}*g(n)\le f(n)\le c_{2}*g(n)$$

- Then we can say that

# Theta Notation



# Properties of Order of Growth

- If $f_{1}(n)$ is order of $g_{1}(n)$ and $f_{2}(n)$ is order of $g_{2}(n)$ then $f_{1}(n)+f_{2}(n)\in O(\max(g_{1}(n),g_{2}(n)))$.
- Polynomials of degree $m\in\Theta(n^{m})$

  That means maximum degree is considered from the polynomial.
- For example: $a1n^{3} + a 2n^{2} + a 3n+c$has the order of growth $\Theta(n^{3})$
- $O(1)<O(\log~n)<O(n)<O(n^{2})<O(2^{n})$.
- Exponential functions $a^{n}$ have different orders of growth for different values of a.

# Best, Average, and Worst-Case Behavior

**Best case Time Complexity:**

- If an algorithm takes minimum amount of time to run to completion for a specific set of input then it is called best case time complexity.
- For example While searching a particular element by using sequential search we get the desired element at first place itself then it is called best case time complexity.

**Worst Case Time Complexity:**

- If an algorithm takes maximum amount of time to run to completion for a specific set of input then it is called worst case time complexity.
- For example - While searching an element by using linear searching method if desired element is placed at the end of the list then we get worst time complexity.

## Average Case Time Complexity:

- The time complexity that we get for certain set of inputs is as a average same.
- Then for corresponding input such a time complexity is called average case time complexity.

## Algorithmic Example:

Consider following algorithm. This algorithm is for sequential search.

Algorithm Seq search (A]0...n-1 ],key)

// Problem Description: This algorithm is for searching the

//key element from an array A[0...n-1] sequentially.

for i=0 to n-1 do if(A[i]=key)then

return i

# Analysis

## Best Case Analysis:

- In above searching algorithm the element key is searched from the list of n elements.
- If the key element is present at first location in the list(A[0...n-1]) then algorithm run for a very short time and thereby we will get the best case time complexity.
- Hence in terms of big-oh notation the time complexity can be denoted as

$$T(n)=O(1)$$

# Worst Case Analysis:

- Worst case time complexity is a time complexity when algorithm runs for a longest time.
- In above searching algorithm the element key is searched from the list of n elements.
- If the key element is present at nth location then clearly the algorithm will run for longest time and thereby we will get the worst case time complexity.
- We can denote the worst case time complexity as

$$T(n) = n$$

- Hence in terms of big-oh notation the time complexity can be denoted as

$$T( n) =O(n)$$

## Average Case Analysis:

- This type of complexity gives information about the behaviour of an algorithm on specific or random input.

# Time Complexity

- The time complexity of an algorithm is the amount of computer time required by an algorithm to run to completion.
- It is difficult to compute the time complexity in terms of phasically clocked time. For instance in multiuser system, executing time depends on many factors such as -
  - System load
  - Number of other programs running
  - Instruction set used
  - Speed of underlying hardware.
- The time complexity is therefore given in terms of frequency count.
- Frequency count is a count denoting number of times of execution of statement.

**For Example**

- If we write a code for calculating sum of n numbers in an array then we can find its time complexity using frequency count. This frequency count denotes how many times the particular statement is executed.