Algorithms

**INTRODUCTION TO ASYMPTOTIC NOTATION**

There will be many solutions to a given problem each of which will be given by an algorithm. Now, we want that algorithm which is going to give me the result quickest while consuming not much of the space (memory). Therefore, design and analysis of algorithms is the subject to design the various algorithms for a problem and analyze them for the best possible algorithm to choose.

Before proceeding, we have some notations to understand. These are called asymptotic notations.

Let’s say we have a function f(n) and as the n increases the rate of growth of time increases in a certain way.



1. **Big-Oh (O)**: Let’s find another function ‘cg(n)’ (above graph) in such a way that the after it gets an input n0 the value of this function is always greater than the f(n). Then if:

If the above satisfies, then we can say that

Which is equivalent to say that f(n) is smaller than g(n).

Let’s take an example: f(n) = 3n + 2 and g(n) = n, is f(n) = O(g(n))?

Note: The tightest bound here is ‘n’. If f(n) = O(g(n)) then definitely n2, n3… will be upper bounds, but we need to see the tightest bound which is ‘n’ in this case.



1. **Big-Omega(Ω)** :



Let’s find another function ‘cg(n)’ (above graph) in such a way that the after it gets an input n0 the value of this function is always smaller than the f(n). Then if:

If the above satisfies, then we can say that

Which is equivalent to say that f(n) is greater than g(n).

*We will see examples about it.*

1. **Big-Theta(Θ):** We say that a function f(n) = Θ(g(n)), if the function f(n) is bounded by both lower and upper bounds. Let’s see the graph for same.



Therefore,

If the above is satisfied then we can say f(n) = Θ(g(n))

Example: f(n) = 3n + 2, g(n) = n. Is f(n) = Θ(g(n))?



**Practical significance of the symbols:**

Given any algorithm:

* O is going to say “worst case time” or the “upper bound”. In other words, if you are giving time complexity in O notation, that will mean “In any case the time will not exceed this”.
* Ω is going to say “best case time” which means in any case you can never achieve better than this.
* Θ is giving you the average case

In practicality we worry about the question “What is the worst case time the algorithm will take for any input?”

For some algorithms, the best and worst case will be same. In those scenarios we go for Θ.