DAA Notes

Unit-1

Big-Theta (Θ) notation

Big theta defines a function's lower and upper bounds, i.e., it exists as both, most, and least boundaries for a given input value.

From the definition : f(n) is Θ(g(n)) if there exists positive numbers c1, c2 and N such that c1g(n) <= f(n) <= c2g(n) for all n >= N.

little-o notation

(definition)

Definition: A theoretical measure of the execution of an algorithm, usually the time or memory needed, given the problem size n, which is usually the number of items. Informally, saying some equation f(n) = o(g(n)) means f(n) becomes insignificant relative to g(n) as n approaches infinity. The notation is read, "f of n is little oh of g of n".

Formal Definition: f(n) = o(g(n)) means for all c > 0 there exists some k > 0 such that 0 ≤ f(n) < cg(n) for all n ≥ k. The value of k must not depend on n, but may depend on c.

Generalization (I am a kind of ...)

big-O notation.

See also ω(n).

Note: As an example, $3n + 4$ is $o(n^2)$ since for any c we can choose $k > (3+ \sqrt{(9+16c)})/2c$. $3n + 4$ is not $o(n)$. $o(f(n))$ is an upper bound, but is not an asymptotically tight bound.

This is lower case "o", not the lower case Greek letter omicron. See the note at big-O notation.

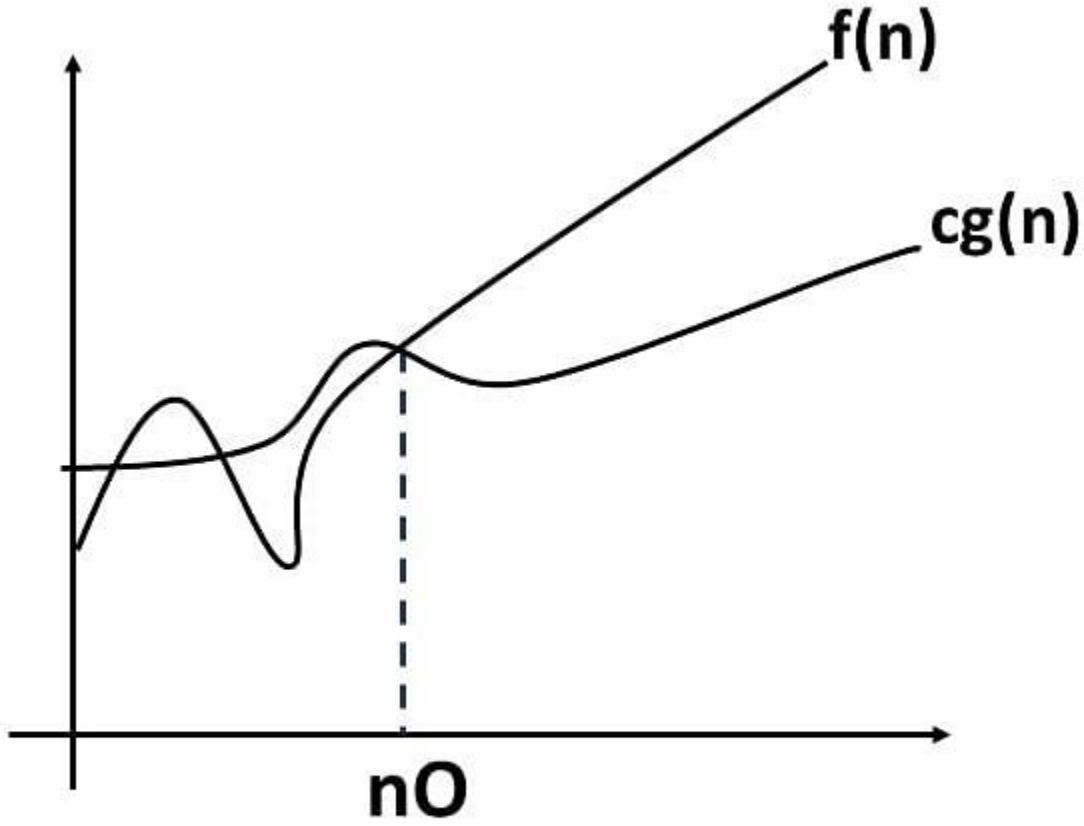
Best Case, Worst Case, and Average Case in Asymptotic Analysis

Best Case: It is defined as the condition that allows an algorithm to complete statement execution in the shortest amount of time. In this case, the execution time serves as a lower bound on the algorithm's time complexity.

Average Case: You add the running times for each possible input combination and take the average in the average case. Here, the execution time serves as both a lower and upper bound on the algorithm's time complexity.

Worst Case: It is defined as the condition that allows an algorithm to complete statement execution in the shortest amount of time possible. In this case, the execution time serves as an upper bound on the algorithm's time complexity.

You will now see how to calculate space and time complexity after grasping the significance of space and time complexity.



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