**TIME COMPLEXITY**

**BUILDING BASIC CONCEPTS**

**DATED : 06/10/2024**

**TIME : 10:18 PM**

**NAME : MUBEEN BUTT**

**ROLL #:BSCS24063**

**DOCUMENTATION #1.**

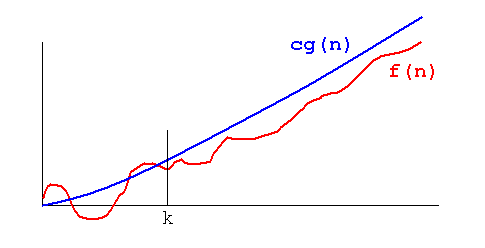
Today I learnt basic functions related to time complexity .I learnt big O notation ,little o notation ,big Omega , little omega , and theta notation .

so what's BIG O?

big O is just upper bound of a function.

lets take a look at formal definition of what BIG O is.

f(n) = O(g(n)) means there are positive constants c and k, such that 0 <= f(n) <= cg(n) for all n >= k. The values of c and k must be fixed for the function f and must not depend on n.



we can also say big O to be asymptotic upper bound of f(x)

As an example , n² + 3n + 4 is O(n²) , since n² + 3n + 4 < 2n² for all n > 10.

why is big O important in time complexity ?

The importance of this measure can be seen in trying to decide whether an algorithm is adequate, but may just need a better implementation, or the algorithm will always be too slow on a big enough input. For instance, quicksort, which is O(n log n) on average, running

on a small desktop computer can beat bubble sort, which is O(n²), running on a supercomputer if there are a lot of numbers to sort. To sort 1,000,000 numbers, the quicksort takes 20,000,000 steps on average, while the bubble sort takes 1,000,000,000,000 steps!

what a little o is ?

its nothing but a strict upper bound of a function . function is less then the upper bound but not equal to.

formal definition:

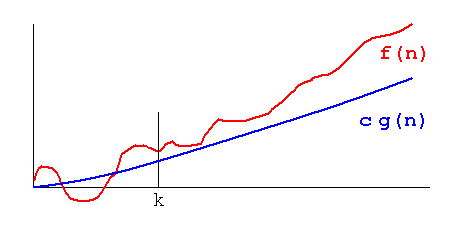
f(n) = o(g(n)) means for all c > 0 there exists some k > 0 such that 0 ≤ f(n) < cg(n) for all n ≥ k. The value of k must not depend on n, but may depend on c.

what a big omega is?

speaking of an informal definition ,big omega is just the lower bound of a function .

formal definition:

f(n) = Ω (g(n)) means there are positive constants c and k, such that 0 ≤ cg(n) ≤ f(n) for all n ≥ k. The values of c and k must be fixed for the function f and must not depend on n.



further more,

This definition is stronger than the traditional mathematical definition. Here, f must be greater than g FOR ALL x bigger than some k. The traditional definition is f is big omega of g if it is not little o of g. That is, one needs only an unbounded sequence of values of

x tending to infinity such that cg(x) ≤ f(x) for all x in the sequence.

so a question arises, what is little omega?

it is noting but but g(n) < f(n) . I mean by that it is strict lower bound of f(n).

formal definition :

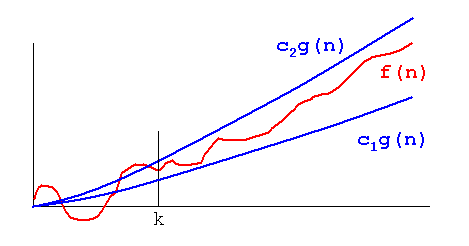
f(n) = ω (g(n)) means that for any positive constant c, there exists a constant k, such that 0 ≤ cg(n) < f(n) for all n ≥ k. The value of k must not depend on n, but may depend on c.

and last but not least , what is theta notation?

speaking of informal definition , theta is just sandwiched between a very close lower and upper bound .

formal definition :

f(n) = Θ (g(n)) means there are positive constants c1, c2, and k, such that 0 ≤ c1g(n) ≤ f(n) ≤ c2g(n) for all n ≥ k. The values of c1, c2, and k must be fixed for the function f and must not depend on n.



Also known as asymptotically tight bound.