Lecture Notes

on

Complexity of an Algorithm

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

09-July 2020

(Be safe and stay at home)

Types of Time Complexity

- **Best-case time complexity:**
  minimum amount of time required to solve a particular problem is called best-case time complexity.
- **Worst-case time complexity:**
  maximum amount of time required to solve a particular problem is called worst-case time complexity.
- **Average-case time complexity:**
  avg. amount of time required to solve a particular problem is called avg.-case time complexity.

Rate of Growth

- The rate at which the running time increases as a function of input, is called rate of growth.
- let us assume, we bought a two items (car & cycle)
  $Total_cost = cost_of_car + cost_of_cycle$
  $Total_cost \approx cost_of_car(Approximation)$

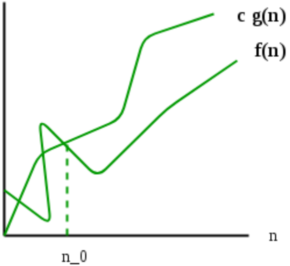
Why it is called Asymptotic Analysis?

- In every case, for a given function $f(n)$, we are trying to find another function $g(n)$ which approximates $f(n)$ at higher values of $'n'$.
- which means, $g(n)$ is also a curve which approximates $f(n)$ at higher values of n.
- In mathematics, we call such curves as asymptotic curves.
- for this reason, we call algorithm analysis as an asysmptotic analysis.

Asymptotic Notations:

1. Big-oh Notation (O)
2. Omega Notation (Ω)
3. theta Notation (θ)
4. small-oh Notation (o)
5. small-omega Notation (ω)

Big-oh Notation (O) (<=)

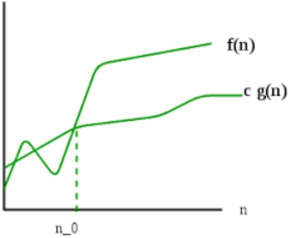


f(n) = O(g(n))

let $f(n) \& g(n)$ be two functions from set of integer.
$f(n) = O(g(n))$ IFF
$f(n) <= c.g(n) : \forall n, n >= n_0$;
c & n_0 are constant.

- Asymptotic upper bound provided by O-notation, which may or may not be aymptotically tight.
  e.g. i) $n^2 = O(n^3)$: not tightest upper bound
  ii) $n^2 = O(n^10)$: not tightest upper bound
  iii) $n^2 = O(n^2)$: tightest upper bound
- $A = O(B)$; where B is either tightes or not-tightest UB.

Omega Notation Ω (>=)

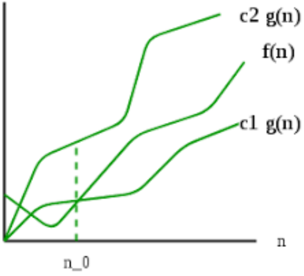


f(n) = Omega(g(n))

let $f(n) \& g(n)$ be two functions from set of integer.
$f(n) = \omega(g(n))$ IFF
$f(n) >= c.g(n) : \forall n, n >= n_0$;
c & n_0 are constant.

- e.g. i) $n^3 = \Omega(n)$: not tightest lower bound
  ii) $n^3 = \Omega(n^2)$: not tightest lower bound
  iii) $n^3 = \Omega(n^3)$: tightest lower bound
- $A = \Omega(B)$; where B is either tightes or not-tightest LB.

theta Notation (θ)



f(n) = theta(g(n))

- $f(n) = \theta(g(n))$ IFF
  i) $f(n) <= c_1.g(n) : \forall n, n >= n_0$;
  ii) $f(n) >= c_2.g(n) : \forall n, n >= n_0$
- $T(A) = O(n^3)$; worst-case (O)
- $T(A) = \omega(n^3)$; best-case(Ω)
- when $worst-case = best-case$; then only θ apply. $T(A) = \theta(n^3)$

small-oh (o) Notation (<):

- we use o-notations to denote an upper bound that is not asymptotically tight.
- i) $n^2 = o(n^3)$: NT
  ii) $n^2 = o(n^{10})$: NT
- $A = o(B)$ where B is not tight upper bound.

small-omega (ω) Notation $(>)$:

- we use ω notations to denote an lower bound that is not asymptotically tight.
- i) $n^3 = \omega(n^2)$: NT
  ii) $n^3 = \omega(n)$: NT
- $A = \omega(B)$ where B is not tight lower bound.

Q & A?

Queries are welcome on slack channel for discussion