-Algorithm analysis- Analysis of complexity of algorithms & finding most efficient algorithm to solve a problem

-Big-O Notation- Statistical measure, used to describe the algorithm complexity.

-- Measured in BEST case, AVERAGE case, & WORST case

-- Always Pick WORST CASE

FROM FASTEST TO SLOWEST

O(1)	constant (or O(k) for constant k)
O(log n)	logarithmic
O(n)	linear
O(n log n)	"n log n"
O(n^2)	quadratic
O(n^3)	cubic
O(n^k)	polynomial (where is k is constant)
O(k^n)	exponential (where constant k > 1)
O(n!)	factorial

TO FIND BIG O NOTATION

1 Find Fastest Growing Term

EX.) T = a*n* + b a*n* = Fastest Growing Term

2 Take out the Coefficient

EX.) a*n* = n -> O(*n*)

EX.2) $T = cn^2 + dn + e$

-- cn^2 = Fastest Growing Term

-- n^2 = No Coefficient

-- $O(n^2)$