CSCI 335 Notes

Anton Goretsky Ralph Vente

Contents

Big O notation

T(n) = O(n^2) bounding the algorithm with the O(n^2) time.

$O(n)$ notation means that the runtime of the algorithm on a dataset of n is bounded above by $O(n)$.

$$T(n) = \Omega(n^2)$$

Omega is the best case runtime.

$$T(n) = \Theta(n^2)$$

Theta is a tight constraint on the runtime.

Alternatively, Big O can be thought of as...

$$T(n) = O(f(N)) \equiv \lim_{N \to \infty}$$

Rules

Rule 1

$T_1(N) = O(f(N))$ and $T_2(N) = O(g(N))$

then

a) $T_1(N) + T_2(N) = O(f(N) + g(N))$

b) $T_1(N)T_2(N) = O(f(N)g(N))$

Rule 2

If $T(n)$ is a polynomial degree k, then $T(N) = \Theta(N^k))$

Rule 3

$$\log^k(N) = O(n)$$

Limits are used to determine the relative growth rate of two functions.