CS310: ALGORITHMS AND DATA STRUCTURES

Transitivity of Asymptotic Notations

- If $T(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$,
- Does this imply that $T(n) = \Theta(h(n))$?

- If $T(n) = O(g(n))$ and $g(n) = O(h(n))$,
- Does this imply that $T(n) = O(h(n))$?

- What about other asymptotic notations?

Reflexivity

- $T(n) = \Theta(T(n))$?
- $T(n) = O(T(n))$?
- $T(n) = \Omega(T(n))$?

Symmetry

- $T(n) = \Theta(f(n)) \Rightarrow f(n) = \Theta(T(n))$?
- $T(n) = \Omega(f(n)) \Rightarrow f(n) = \Omega(T(n))$?
- $T(n) = O(f(n)) \Rightarrow f(n) = O(T(n))$?
- What about o and ω?

Other Properties

- If $f(n) = O(g(n))$, then $g(n) = \Omega(f(n))$ and vice versa
- If $f(n) = o(g(n))$, then $g(n) = \omega(f(n))$ and vice versa

Review

- What properties are satisfied by the following asymptotic notations?
  - Θ:
  - Ω:
  - O:
  - ω:
  - o:

Parallels between Asymptotic Notations and Arithmetic Operators

- $T(n) = \Theta(f(n)) \Leftrightarrow t = f$
- $T(n) = \Omega(f(n)) \Leftrightarrow t \geq f$
- $T(n) = O(f(n)) \Leftrightarrow t \leq f$
- $T(n) = \omega(f(n)) \Leftrightarrow t > f$
- $T(n) = o(f(n)) \Leftrightarrow t < f$