Design and Analysis of Algorithm

Department of Computer Science and Engineering
National Institute of Technology Raipur

Pramod Mane

January 16, 2021

Outline

Outline

Comparing Asymptotic Functions

Transitivity

$f(n) = \mathcal{O}(g(n))$ and $g(n) = \mathcal{O}(h(n))$ implies $f(n) = \mathcal{O}(h(n))$

Proof.

$f(n) = \mathcal{O}(g(n)) and g(n) = \mathcal{O}(h(n)) \Rightarrow f(n) = \mathcal{O}(h(n))$

By the definition of Big-Oh($\mathcal{O}$), there exists positive constants c, n_0 such that $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$

$\Rightarrow f(n) \leq c_1 \cdot g(n)$

$\Rightarrow g(n) \leq c_2 \cdot h(n)$

$\Rightarrow f(n) \leq c_1 \cdot c_2 h(n)$

$\Rightarrow f(n) \leq c \cdot h(n)$, where, $c = c_1 \cdot c_2$.

By the definition, $f(n) = \mathcal{O}(h(n))$. $\square$

Comparing Asymptotic Functions

Transitivity

- $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$ implies $f(n) = \Theta(h(n))$,
- $f(n) = \Omega(g(n))$ and $g(n) = \Omega(h(n))$ implies $f(n) = \Omega(h(n))$,
- $f(n) = o(g(n))$ and $g(n) = o(h(n))$ implies $f(n) = o(h(n))$,
- $f(n) = \omega(g(n))$ and $g(n) = \omega(h(n))$ implies $f(n) = \omega(h(n))$.

Comparing Asymptotic Functions Cont...

Reflexivity

- $f(n) = \Theta(f(n))$,
- $f(n) = \mathcal{O}(f(n))$,
- $f(n) = \Omega(f(n))$.

Comparing Asymptotic Functions Cont...

Symmetry

- $f(n) = \Theta(g(n))$ if and only if $g(n) = \Theta(f(n))$.

Proof.

Necessary part:
$f(n) = \Theta(g(n)) \Rightarrow g(n) = \Theta(f(n))$
By the definition of Θ, there exists positive constants c_1, c_2, n_0 such that $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ for all $n \geq n_0$
$\Rightarrow g(n) \leq (\frac{1}{c_1}) \cdot f(n) and g(n) \geq (\frac{1}{c_2}) \cdot f(n)$
$\Rightarrow (\frac{1}{c_2}) \cdot f(n) \leq g(n) \leq (\frac{1}{c_1}) \cdot f(n)$
Since c_1 and c_2 are positive constants, $\frac{1}{c_1}$ and $\frac{1}{c_2}$ are well defined.
Therefore, by the definition of $\Theta, g(n) = \Theta(f(n))$ $\square$

Comparing Asymptotic Functions Cont...

Sufficiency part:

$g(n) = \Theta(f(n)) \Rightarrow f(n) = \Theta(g(n))$

By the definition of Θ, there exists positive constants c_1, c_2, n_0 such that $c_1 \cdot f(n) \leq g(n) \leq c_2 \cdot f(n)$ for all $n \geq n_0$

$\Rightarrow f(n) \leq (\frac{1}{c_1}) \cdot g(n)$ and $f(n) \geq (\frac{1}{c_2}) \cdot g(n)$

$\Rightarrow (\frac{1}{c_2}) \cdot g(n) \leq f(n) \leq (\frac{1}{c_1}) \cdot g(n)$

By the definition of $\Theta, f(n) = \Theta(g(n))$

Comparing Asymptotic Functions Cont...

Transpose Symmetry

- $f(n) = \Theta(g(n))$ if and only if $g(n) = \Theta(f(n))$.

Proof.

Necessary part:
$f(n) = \mathcal{O}(g(n)) \Rightarrow g(n) = \Theta(f(n))$
By the definition of Big-Oh $(\mathcal{O}) \Rightarrow f(n) \leq c \cdot g(n)$ for some positive constant $c \Rightarrow g(n) \geq (\frac{1}{c}) \cdot f(n)$
By the definition of Omega $(\Theta), g(n) = \Theta(f(n))$
Sufficiency part:
$g(n) = \Theta(f(n)) \Rightarrow f(n) = O(g(n))$
By the definition of Omega (Θ), for some positive constant $c \Rightarrow g(n) \geq c \cdot f(n) \Rightarrow f(n) \leq (\frac{1}{c}) \cdot g(n)$
By the definition of Big-Oh $(\mathcal{O}), f(n) = \mathcal{O}(g(n))$.

□

Comparing Asymptotic Functions Cont...

Since these properties hold for asymptotic notations, analogies can be drawn between functions f(n) and g(n) and two real numbers a and b.

- $g(n) = \mathcal{O}(f(n)) is similar to a \leq b,$
- $g(n) = \Omega(f(n)) is similar to a \geq b,$
- $g(n) = \Theta(f(n)) is similar to a = b,$
- $g(n) = o(f(n)) is similar to a < b,$
- $g(n) = \omega(f(n)) is similar to a > b.$

Thank You!