INFORMATION & COMPUTER SCIENCES
UNIVERSITY of HAWAIʻI at MĀNOA

Ch 3.2: The Growth of Functions

ICS 141: Discrete Mathematics for Computer Science I

KYLE BERNEY
DEPARTMENT OF ICS, UNIVERSITY OF HAWAII AT MANOA

Asymptotic Analysis

- When analyzing algorithms, many architecture specific parameters determine the overall runtime
  - The number of cycles needed to perform specific operations
- Ex: The runtime of an algorithm on a supercomputer will be different than the runtime of the same algorithm execute on a personal computer (PC)
- Want to study the runtime of algorithms without worrying about specific architectural dependent constants

Asymptotic Analysis

- Definition: Asymptotic analysis is a method for describing the behavior of functions as the input size grows “large”.
  - Multiplicative constants and lower-order terms are dominated by the effects of the input size
- Typically, an algorithm that is asymptotically more efficent will be the best choice
  - There may be better choices for “small” inputs

Asymptotic Notation

- In this section, we will introduce various asymptotic notations
- The different asymptotic bounds we will use are analogous to equality and inequality relations:
  - $O \approx \leq$
  - $\Omega \approx \geq$
  - $\Theta \approx =$
  - $o \approx <$
  - $\omega \approx >$

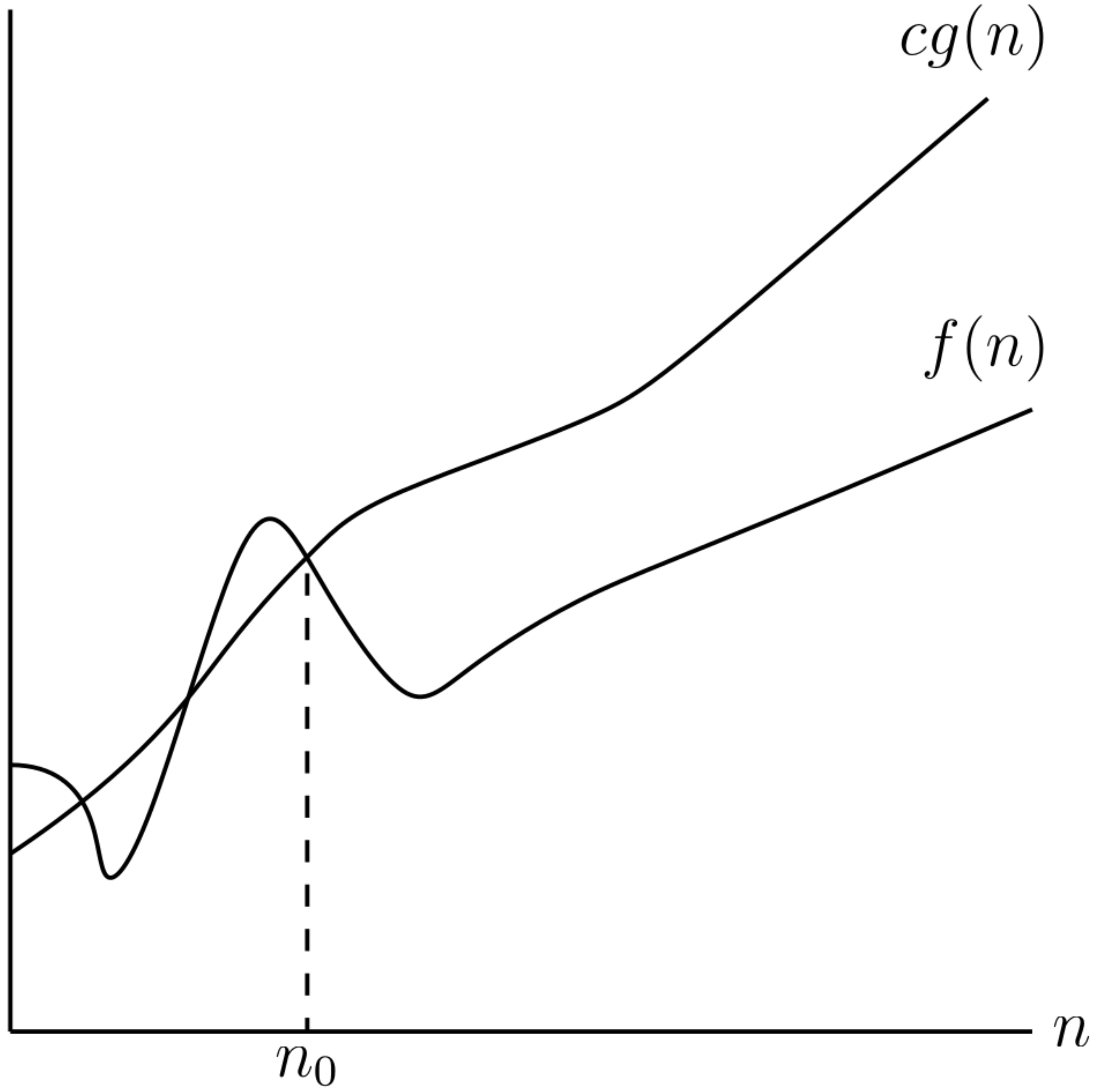
Big-O Notation

- Definition: The asymptotic upper bound of a function $g(n)$, denoted $O(g(n))$, is the set of functions

$$O(g(n)) = \{f(n) : \text{there exists positive constants } c \text{ and } n_0 \text{ such that } 0 \leq f(n) \leq cg(n) \text{ for all } n \geq n_0\}$$

- Read as "Big-Oh of g of n"
- Write $f(n) = O(g(n))$ to indicate that a function $f(n)$ is a member of the set $O(g(n))$

Big-O Notation



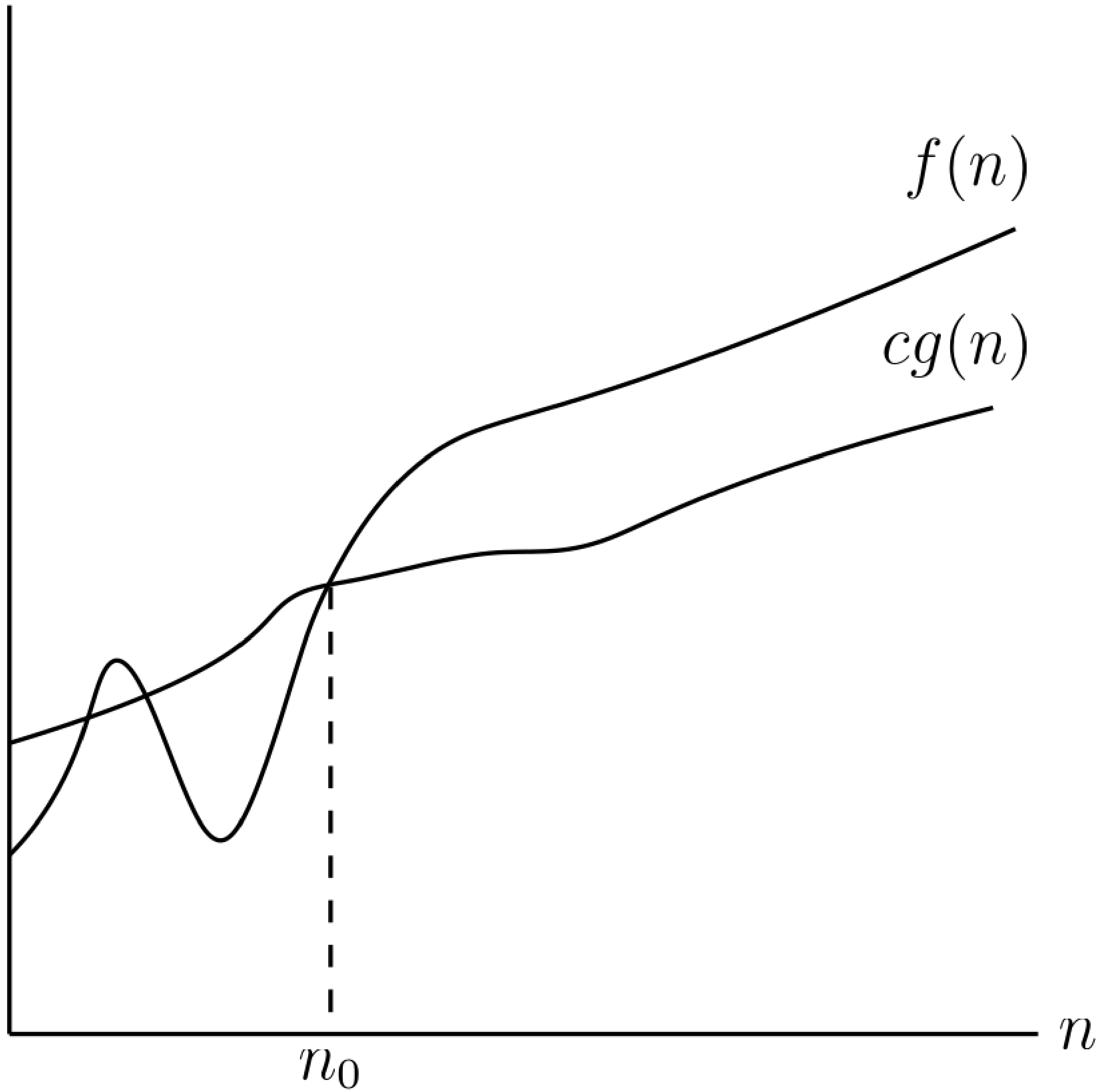
Ω Notation

- Definition: The asymptotic lower bound of a function $g(n)$, denoted $\Omega(g(n))$, is the set of functions

$$\Omega(g(n)) = \{f(n) : \text{there exists positive constants } c \text{ and } n_0 \text{ such that } 0 \leq cg(n) \leq f(n) \text{ for all } n \geq n_0\}$$

- Read as "Omega of *g* of *n*"
- Write $f(n) = \Omega(g(n))$ to indicate that a function $f(n)$ is a member of the set $\Omega(g(n))$

Ω Notation



Θ Notation

- Definition: The asymptotically tight bound of a function $g(n)$, denoted $\Theta(g(n))$, is the set of functions

$$\Theta(g(n)) = \{f(n) : \text{there exists positive constants } c_1, c_2, \text{ and } n_0 \text{ such that } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \text{ for all } n \geq n_0\}$$

- Read as "Theta of g of n"
- Write $f(n) = \Theta(g(n))$ to indicate that a function $f(n)$ is a member of the set $\Theta(g(n))$

Θ Notation

- Theorem: For any two function $f(n)$ and $g(n)$, we have

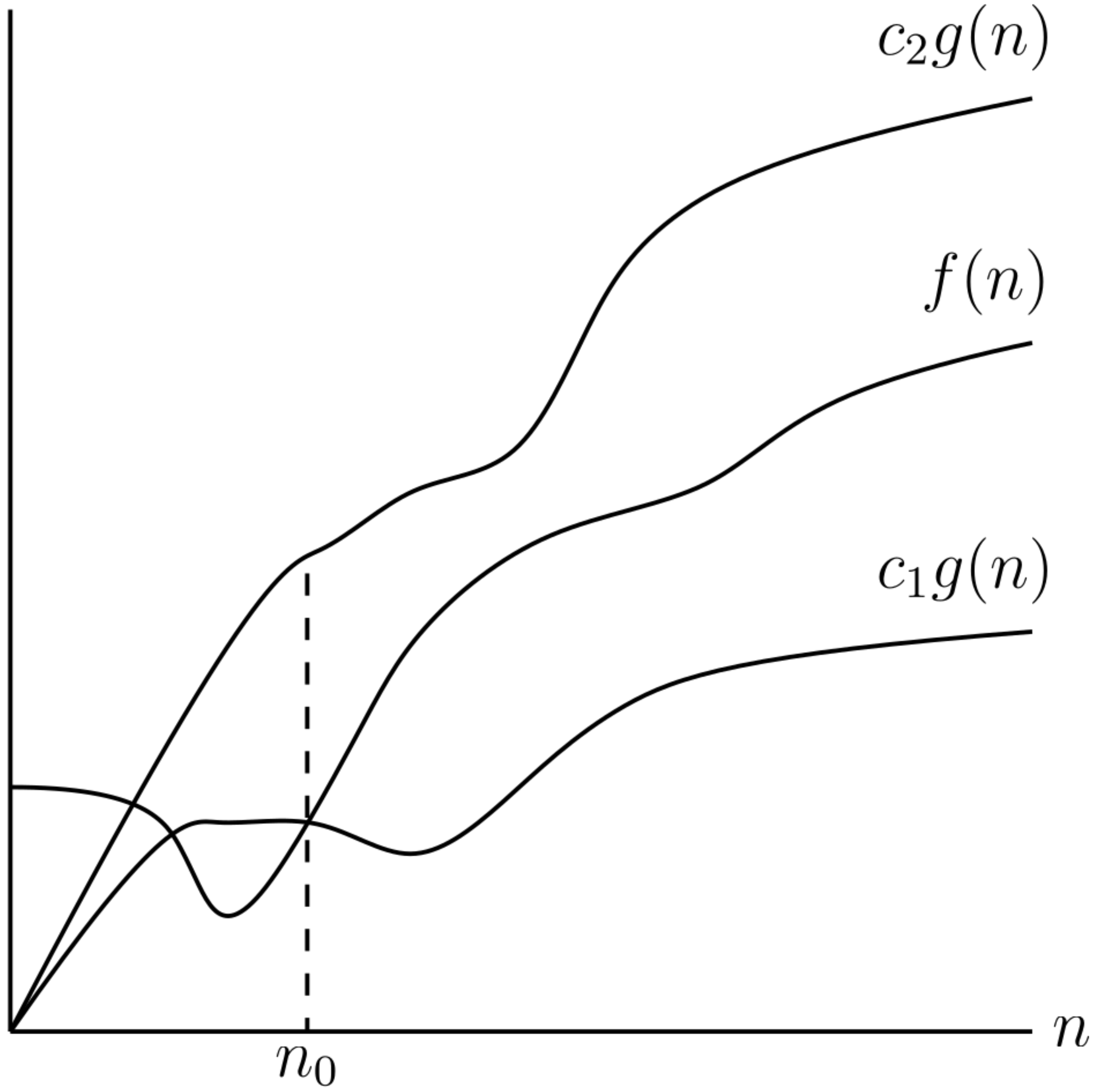
$$f(n) = \Theta(g(n))$$

if and only if

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

$$\text{and } f(n) = \Omega(g(n))$$

Θ Notation



o Notation

- The O and Ω bounds may or may not be asymptotically tight
  - $2n^2 = O(n^2)$ is asymptotically tight
  - $2n = O(n^2)$ is not asymptotically tight
- Definition:

$$o(g(n)) = \{f(n) : \text{for any positive constants } c \text{ there exists a positive constant } n_0 \text{ such that } 0 \leq f(n) < cg(n) \text{ for all } n \geq n_0\}$$

- Read as "Little-oh of g of n"
- Write $f(n) = o(g(n))$ to indicate that a function $f(n)$ is a member of the set $o(g(n))$
- Never asymptotically tight

ω Notation

- The O and Ω bounds may or may not be asymptotically tight
  - $2n^2 = \Omega(n^2)$ is asymptotically tight
  - $2n^3 = \Omega(n^2)$ is not asymptotically tight
- Definition:

$$\omega(g(n)) = \{f(n) : \text{for any positive constants } c \text{ there exists a positive constant } n_0 \text{ such that } 0 \leq cg(n) < f(n) \text{ for all } n \geq n_0\}$$

- Read as "Little-omega of g of n"
- Write $f(n) = \omega(g(n))$ to indicate that a function $f(n)$ is a member of the set $\omega(g(n))$
- Never asymptotically tight

Comparing Functions

- Transitivity:
  - If $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$ $\Rightarrow f(n) = \Theta(h(n))$
  - If $f(n) = O(g(n))$ and $g(n) = O(h(n))$ $\Rightarrow f(n) = O(h(n))$
  - If $f(n) = \Omega(g(n))$ and $g(n) = \Omega(h(n))$ $\Rightarrow f(n) = \Omega(h(n))$
  - If $f(n) = o(g(n))$ and $g(n) = o(h(n))$ $\Rightarrow f(n) = o(h(n))$
  - If $f(n) = \omega(g(n))$ and $g(n) = \omega(h(n))$ $\Rightarrow f(n) = \omega(h(n))$

Comparing Functions

- Reflexivity:
  - If $f(n) = \Theta(f(n))$
  - If $f(n) = O(f(n))$
  - If $f(n) = \Omega(f(n))$

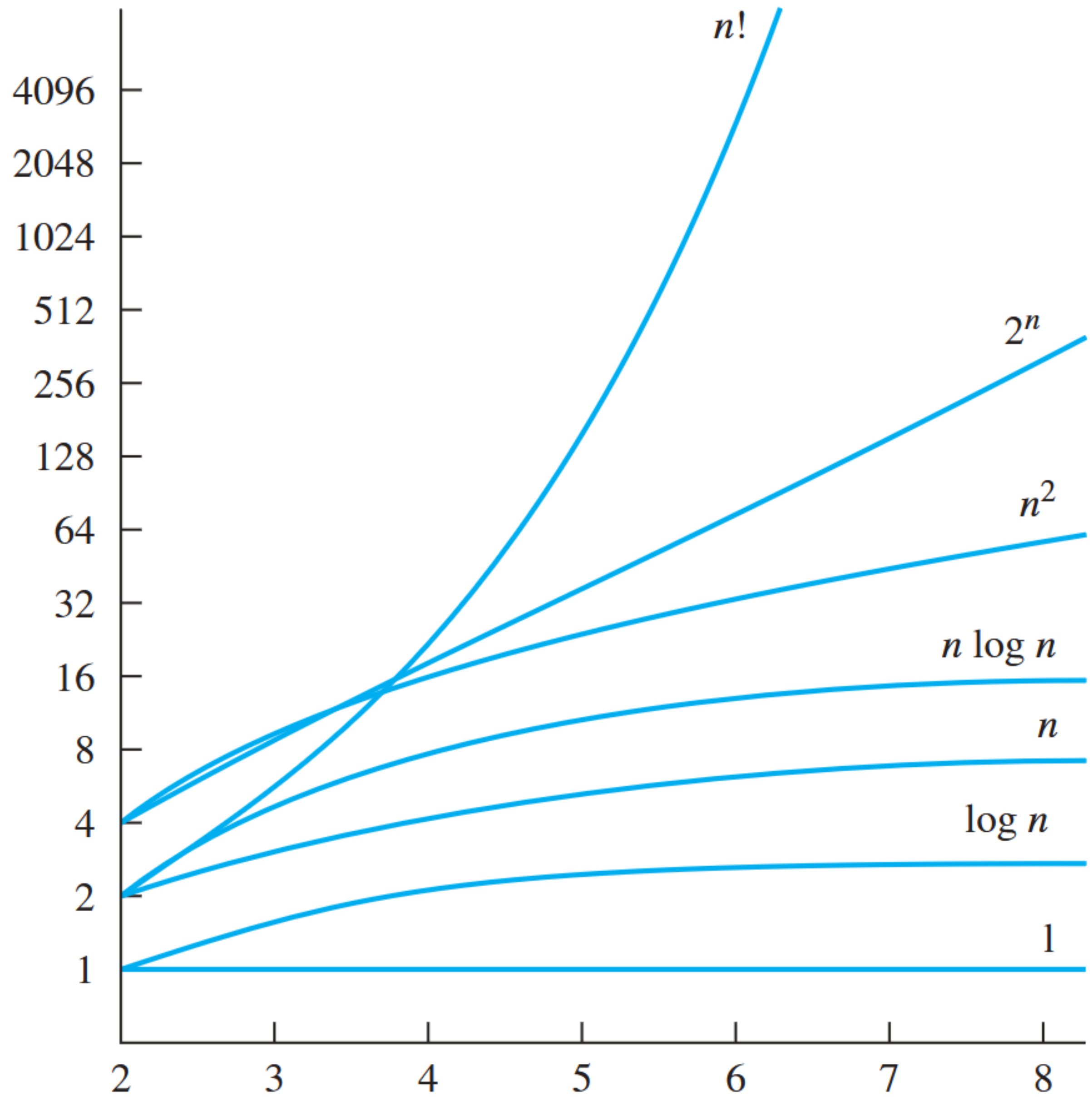
Comparing Functions

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

Comparing Functions

- Transpose Symmetry:
  - $f(n) = O(g(n))$ if and only if $g(n) = \Omega(f(n))$
  - $f(n) = o(g(n))$ if and only if $g(n) = \omega(f(n))$

Common Functions and Useful Facts



Common Functions and Useful Facts

- Exponentials:
  - For all real constants a and b, $n^b = o(a^n)$
  - In other words, any exponential function greater than 1 grows faster than any polynomial function
- Logarithms:
  - For $a > 0$, $(\log n)^b = o(n^a)$
  - In other words, any positive polynomial function grows faster than any polylogarithmic function
- Factorials:
  - $n! = \omega(2^n)$
  - $n! = o(n^n)$
  - $\log(n!) = \Theta(n \log n)$