CS 421: Design and Analysis of Algorithms

Chapter 3: Growth of Functions

Dr. Gaby Dagher
Department of Computer Science
Boise State University

January 20, 2022

BOISE STATE UNIVERSITY

Content of this Chapter

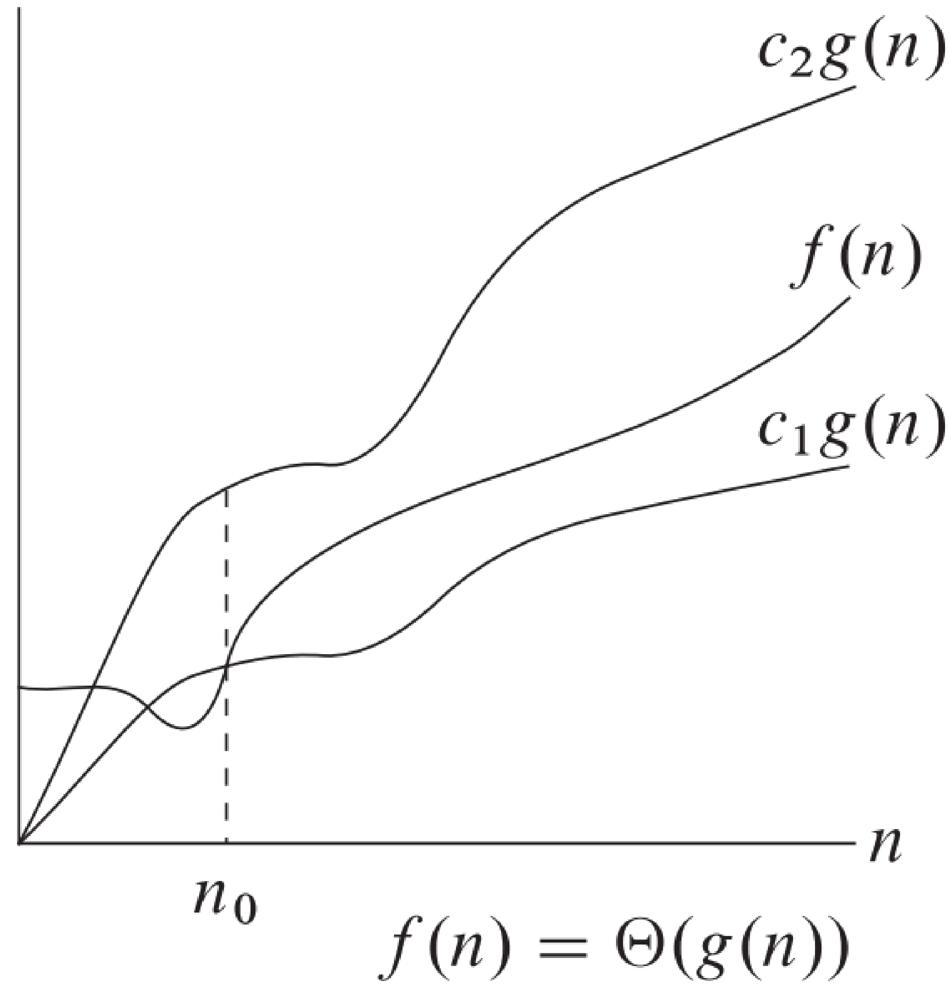
Asymptotic notations:

- **Θ–notation**
- O-notation
- Ω–notation
- Other notations
- Comparing functions
- Comparing Growth Rates Using Limits

Asymptotic Efficiency

- When we look at input sizes large enough to make only the *order of growth* of the running time relevant, we are studying the *asymptotic efficiency of algorithms*.

Θ-notation (tight bounds)



Θ-notation (tight bounds)

We write $f(n) = \Theta(g(n))$ if there exist constants $c_1, c_2, n_0 > 0$ such that:
$0 \leq c_1.g(n) \leq f(n) \leq c_2.g(n)$ for all $n \geq n_0$

- $f(n) \in \Theta(g(n))$
- $g(n)$ is an *asymptotically tight bound* for $f(n)$

Θ-notation (tight bounds)

EXAMPLE #1:

$$\tfrac{1}{2}n^2 - 3n = \Theta(n^2)$$ **True/False?**

EXAMPLE #2:

$$n^3 = \Theta(n^2)$$ **True/False?**

EXAMPLE #3:

$$n^3 + 4n - 6 = \Theta(n^3)$$ **True/False?**

Θ-notation (tight bounds)

Note#1: Often, c_1 and/or c_2 are less than 1.

Note#2: If *f(n)* is *polynomial* and the coefficients of *f(n)* are all *positive*, then:

- c_1 = the coefficient of the highest order term in *f(n)*
- c_2 = the sum of all coefficients of *f(n)*
- $n_0 = 1$

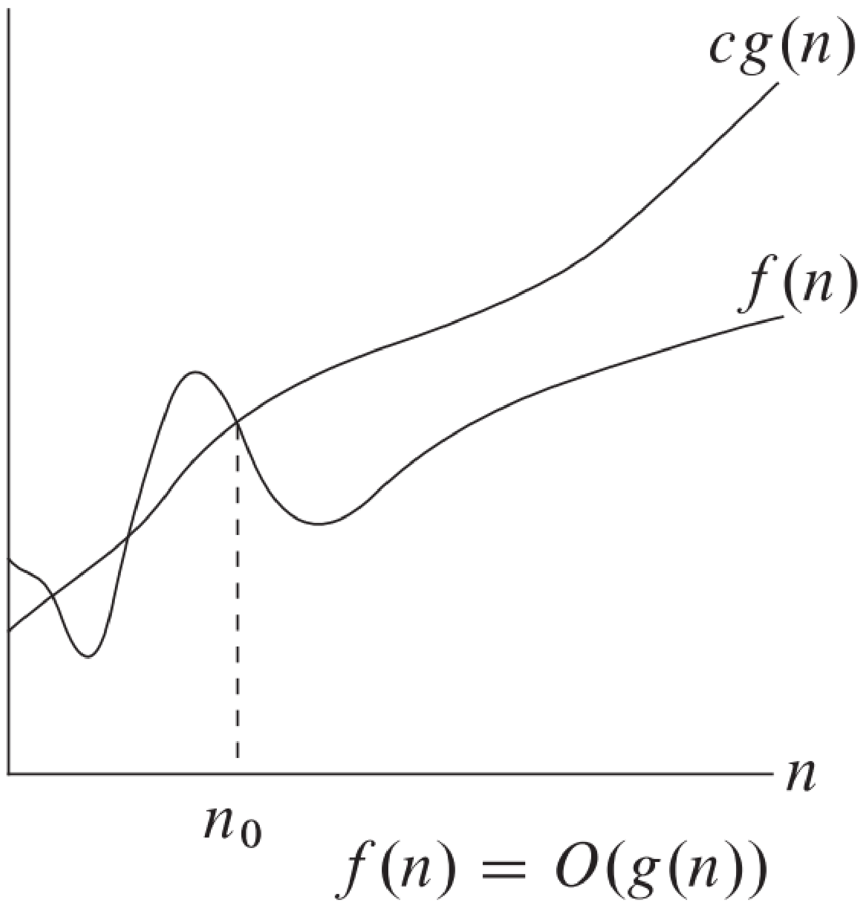
EXAMPLE: $n^3 + 4n + 6 = \Theta(n^3)$

Content of this Chapter

Asymptotic notations:

- Θ–notation
- **O-notation**
- Ω–notation
- Other notations
- Comparing functions
- Comparing Growth Rates Using Limits

O-notation (upper bounds)



O-notation (upper bounds)

We write $f(n) = O(g(n))$ if there exist constants $c > 0$, $n_0 > 0$ such that:
$0 \leq f(n) \leq c.g(n)$ for all $n \geq n_0$

- $f(n) \in O(g(n))$
- $g(n)$ is an *asymptotic upper bound* for $f(n)$

O-notation (upper bounds)

EXAMPLE: $2n^2 = O(n^3)$ $(c = 1, n_0 = 2)$

<u>Rule</u>: If $a \le b$ then $n^a \le n^b$: $n \ge 1$.

<u>Strategy</u>: sometimes, c could be the sum of the *positive* coefficients of $f(n)$.

EXAMPLE: $n^2 + 4 = O(n^2)$ $(c = 5, n_0 = 1)$

$n^2 - 3n + 4 = O(n^2)$ $(c = ?, n_0 = ?)$

Θ and O **notations**

- O-notation describes an upper bound.

- When O-notation is used to bound the worst-case running time of an algorithm, we have a *blanket statement* (for every input).

- Example: **Insertion Sort**
  - O(n^2) bound on worst-case running time. *Is it a blanket statement?*
    Yes, because it applies to its running time on every input.
  - Θ(n^2) bound on worst-case running time. *Is it a blanket statement?*
    No, because it does not apply to its running time on every input. More specifically, when the input is already sorted, insertion sort runs in Θ(n) time.

Macro substitution

Convention: A set in a formula represents an anonymous function in the set.

EXAMPLE: $f(n) = n^3 + O(n^2)$

means

$f(n) = n^3 + h(n)$

for some $h(n) \in O(n^2)$.

Macro substitution

Convention: A set in a formula represents an anonymous function in the set.

EXAMPLE: $n^2 + O(n) = O(n^2)$

means

for any $f(n) \in O(n)$:

$$n^2 + f(n) = h(n)$$

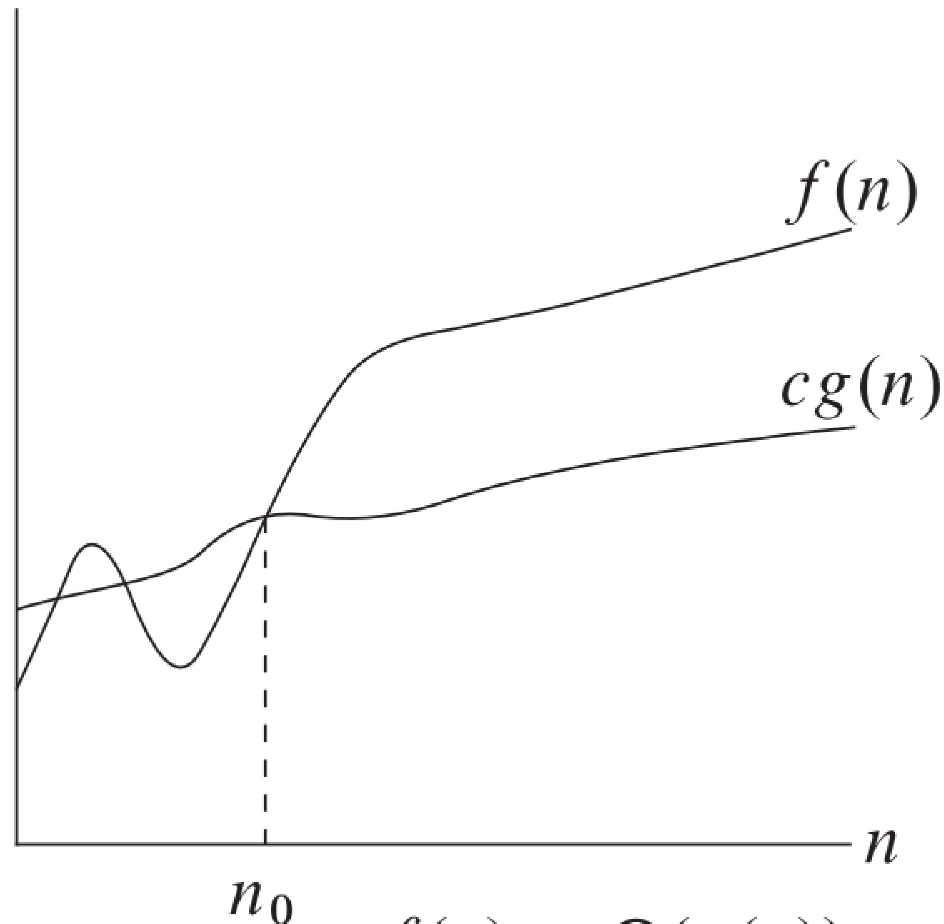
for some $h(n) \in O(n^2)$.

Content of this Chapter

Asymptotic notations:

- Θ–notation
- O-notation
- **Ω–notation**
- Other notations
- Comparing functions
- Comparing Growth Rates Using Limits

Ω-notation (lower bounds)



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

Ω-notation (lower bounds)

$\Omega(g(n)) = \{ f(n)$: there exist constants
$c > 0, n_0 > 0$ such that:
$0 \le c.g(n) \le f(n)$ for all $n \ge n_0$ $\}$

Ω-notation (lower bounds)

- $f(n) = \Omega(g(n))$ (no qualifier), means a *lower bound on the best-case* running time of $f(n)$.
- For example:
  - The best-case running time of insertion sort is $\Omega(n)$.
  - This implies that the running time of insertion sort is $\Omega(n)$.
- However:
  - Insertion sort is not $\Omega(n^2)$, since there exists an input for which it runs in $\Omega(n)$ time (e.g., input is already sorted).
  - Insertion sort is *worst-case* $\Omega(n^2)$, since there exists an input for which it runs in $\Omega(n^2)$ time (e.g., input is sorted in reverse order).

Ω-notation (lower bounds)

EXAMPLE: $n^3 + 3n^2 = \Omega(n^2)$ $(c = ?, n_0 = ?)$

Solution:

$0 \le c.g(n) \le f(n) \rightarrow c.n^2 \le n^3 + 3n^2$

$\rightarrow c \le n + 3$

$\rightarrow$ $c = 1$ and $n_0 = 1$

Ω-notation (lower bounds)

EXAMPLE: $\sqrt{n} = \Omega(\lg n) \quad (c = ?, n_0 = ?)$

Solution:

$0 \le c.g(n) \le f(n) \rightarrow c.lg(n) \le n^{1/2}$

$\rightarrow c \le n^{1/2}/\,lg(n)$

n	1	2	3	**4**	5	6	7	8	9	10	11	12	13	14	15	**16**	17	18	19	20
SQRT(n)	1	1.414	1.732	2	2.236	2.449	2.646	2.828	3	3.162	3.317	3.464	3.606	3.742	3.873	4	4.123	4.243	4.359	4.472
lg n	0	1	1.585	2	2.322	2.585	2.807	3	3.17	3.322	3.459	3.585	3.7	3.807	3.907	4	4.087	4.17	4.248	4.322
SQRT(n) / lg n	#####	1.414	1.093	**1**	0.963	0.948	0.942	0.943	0.946	0.952	0.959	0.966	0.974	0.983	0.991	**1**	1.009	1.017	1.026	1.035

$\rightarrow$ $c = 1$ and $n_0 = 16$

Ω-notation (lower bounds)

EXAMPLE: $n \lg n - 2n + 11 = \Omega(n \lg n)$

<u>Solution</u>:

Observe that: $n \lg n - 2n \leq n \lg n - 2n + 11$ → we will consider $f(n) = n \lg n - 2n$ going forward.

$0 \leq c.g(n) \leq f(n)$ → $c.n \lg n \leq n \lg n - 2n$

→ $c \leq 1 - 2/\lg n$

If $n = 5$ → $1 - 2/\lg n = 0.861$

→ $c = 0.5$ and $n_0 = 5$

Θ, O, Ω notations

$$\boxed{f(n) = \Theta(g(n)) \rightarrow f(n) = O(g(n))}$$

$$\Theta(g(n)) \subseteq O(g(n))$$

$$\boxed{f(n) = \Theta(g(n)) \rightarrow f(n) = \Omega(g(n))}$$

$$\Theta(g(n)) \subseteq \Omega(g(n))$$

$$\boxed{\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))}$$

Content of this Chapter

Asymptotic notations:

- Θ–notation
- O-notation
- Ω–notation
- **Other notations**
- Comparing functions
- Comparing Growth Rates Using Limits

o-notation

O-notation and Ω-notation are like ≤ and ≥.
o-notation and ω-notation are like < and >.

$o(g(n)) = \{f(n) :$ for __any__ $c>0$, there is a constant $n_0 > 0$ such that $0 \leq f(n) < c.g(n)$ for all $n \geq n_0\}$

EXAMPLE: $2n^2 = o(n^3)$ $\quad (n_0 = ?)$

Answer: $n_0 > 2/c$ $\quad$ *e.g.* $n_0 = \lceil 1 + 2/c \rceil$

ω-notation

O-notation and Ω-notation are like ≤ and ≥.
o-notation and ω-notation are like < and >.

$\omega(g(n)) = \{f(n) :$ for <u>any</u> $c>0$, there is a constant $n_0>0$ such that $0 \leq c.g(n) < f(n)$ for all $n \geq n_0$

Content of this Chapter

Asymptotic notations:

- Θ–notation
- O-notation
- Ω–notation
- Other notations

➢ **Comparing functions**

- Comparing Growth Rates Using Limits

Comparing Functions

Transitivity:

$$
\begin{aligned}
f(n) &= \Theta(g(n)) \text{ and } g(n) = \Theta(h(n)) &&\text{imply} && f(n) = \Theta(h(n)) \,, \\
f(n) &= O(g(n)) \text{ and } g(n) = O(h(n)) &&\text{imply} && f(n) = O(h(n)) \,, \\
f(n) &= \Omega(g(n)) \text{ and } g(n) = \Omega(h(n)) &&\text{imply} && f(n) = \Omega(h(n)) \,, \\
f(n) &= o(g(n)) \text{ and } g(n) = o(h(n)) &&\text{imply} && f(n) = o(h(n)) \,, \\
f(n) &= \omega(g(n)) \text{ and } g(n) = \omega(h(n)) &&\text{imply} && f(n) = \omega(h(n)) \,.
\end{aligned}
$$

Reflexivity:

$$
\begin{aligned}
f(n) &= \Theta(f(n)) \,, \\
f(n) &= O(f(n)) \,, \\
f(n) &= \Omega(f(n)) \,.
\end{aligned}
$$

Comparing Functions

Symmetry:

$$f(n) = \Theta(g(n)) \text{ if and only if } g(n) = \Theta(f(n)) .$$

Transpose symmetry:

$$f(n) = O(g(n)) \text{ if and only if } g(n) = \Omega(f(n)) ,$$
$$f(n) = o(g(n)) \text{ if and only if } g(n) = \omega(f(n)) .$$

Comparing Functions

Comparison of two functions *f* and g and the comparison of two real numbers *a* and *b*:

$$
\begin{array}{lll}
f(n) = O(g(n)) & \text{is like} & a \le b \,, \\
f(n) = \Omega(g(n)) & \text{is like} & a \ge b \,, \\
f(n) = \Theta(g(n)) & \text{is like} & a = b \,, \\
f(n) = o(g(n)) & \text{is like} & a < b \,, \\
f(n) = \omega(g(n)) & \text{is like} & a > b \,.
\end{array}
$$

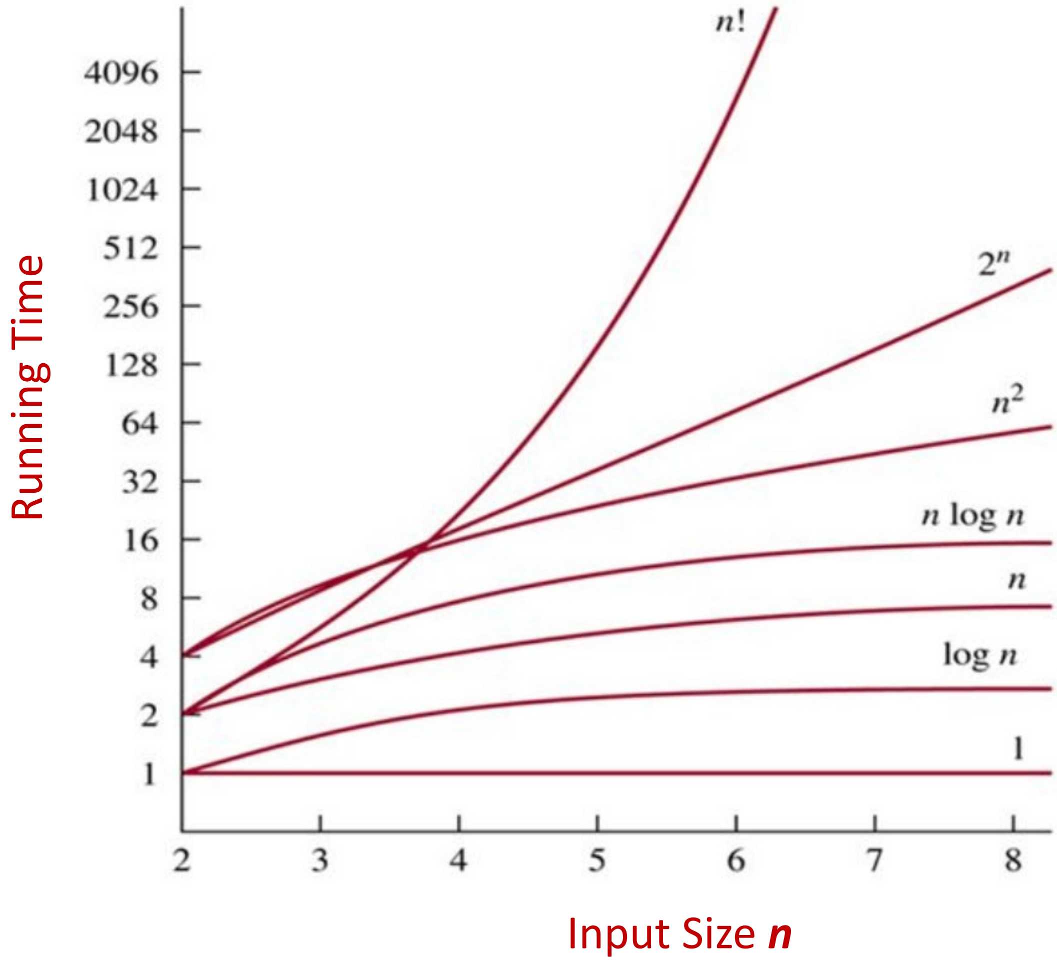
Analogy does not apply:

Trichotomy: For any two real numbers a and b, exactly one of the following must hold: $a < b$, $a = b$, or $a > b$.

Growth of Common Functions

Class	Name	Examples
1	Constant	Only used in best-case efficiencies.
log n	logarithmic	Result of cutting problem size by a constant factor, like Binary Search
n	linear	Algorithms that scan a list of size *n*, like Sequential, or Linear, Search
n log n	n-log-n	Divide-and-Conquer algorithms, like Merge Sort and Quick Sort
n^2	quadratic	Efficiencies with two embedded loops, Bubble Sort and Insertion Sort
n^3	cubic	Efficiencies with three embedded loops, like many linear algebra algorithms
2^n	exponential	Algorithms that generate all sub-sets of an *n*-element set
n!	factorial	Algorithms that generate all permutations of an *n*-element set

Growth of Common Functions



Content of this Chapter

Asymptotic notations:

- Θ–notation
- O-notation
- Ω–notation
- Other notations
- Comparing functions

➢ **Comparing Growth Rates Using Limits**

Comparing Growth Rates Using Limits

Case I: $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

- *f(n)* has <u>smaller</u> growth rate than *g(n)*
- $f(n) = o(g(n))$ → $f(n) = O(g(n))$

Case II: $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

- *f(n)* has <u>larger</u> growth rate than *g(n)*
- $f(n) = \omega(g(n))$ → $f(n) = \Omega(g(n))$

Case III: $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \text{ c (constant)}$

- *f(n)* has <u>same</u> growth rate as *g(n)*
- $f(n) = \Theta(g(n))$ → $f(n) = O(g(n))$ and
  $f(n) = \Omega(g(n))$

Comparing Growth Rates Using Limits

EXAMPLE: $5n^2 - 2n + 8 \stackrel{?}{=} O(n^2)$

Solution:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{5n^2 - 2n + 8}{n^2}$$

$$= \lim_{n\to\infty} 5 + \lim_{n\to\infty} \frac{-2}{n} + \lim_{n\to\infty} \frac{8}{n^2} = 5 \quad \boxed{\text{Case III}}$$

$$\Longrightarrow 5n^2 - 2n + 8 = \Theta(n^2) \Longrightarrow \boxed{5n^2 - 2n + 8 = O(n^2)}$$

Comparing Growth Rates Using Limits

Example: $n \log n \overset{?}{=} \Theta(n^2)$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n \log n}{n^2} = \lim_{n\to\infty} \frac{\log n}{n} = \frac{\infty}{\infty} \quad \boxed{\text{Undefined!}}$$

Recall that: $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = \lim\limits_{n\to\infty} \frac{f'(n)}{g'(n)}$ (L'Hôpital's rule)

$$\Longrightarrow \lim_{n\to\infty} \frac{\log n}{n} = \lim_{n\to\infty} \frac{1/(n \,.\, \ln 10)}{1} = \lim_{n\to\infty} \frac{1}{n \,.\, \ln 10} = 0$$

$$\boxed{\frac{d}{dn} \log_a n = \frac{1}{n \ln a}}$$

$$\Longrightarrow \boxed{\text{Case I}} \Longrightarrow n \log n = o(n^2) \Longrightarrow \boxed{n \log n \neq \Theta(n^2)}$$

Comparing Growth Rates Using Limits

EXAMPLE: $n^3 + 3n^2 - 5n - 1 \overset{?}{=} \Omega(n)$

Solution:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^3 + 3n^2 - 5n - 1}{n}$$

$$= \lim_{n\to\infty} n^2 + \lim_{n\to\infty} 3n + \lim_{n\to\infty} -5 + \lim_{n\to\infty} \frac{-1}{n} = \infty \quad \boxed{\text{Case II}}$$

$$\Longrightarrow n^3+3n^2-5n-1 = \omega(n) \Longrightarrow \boxed{n^3+3n^2-5n-1 = \Omega(n)}$$