CSEN 703 - Analysis and Design of Algorithms

Lecture 2 - Asymptotic Analysis

Dr. Nourhan Ehab
nourhan.ehab@guc.edu.eg

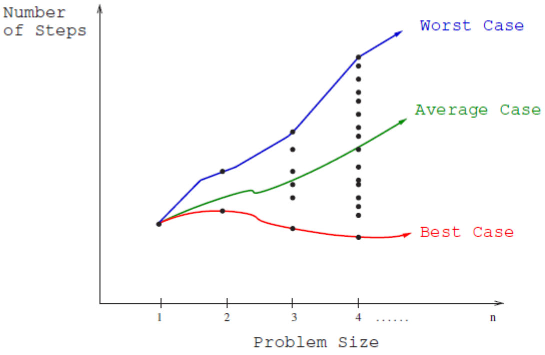
Department of Computer Science and Engineering
Faculty of Media Engineering and Technology

Outline

1. Motivation
2. Asymptotic Notations
3. Recap

In the Previous Lecture

- Analyzing algorithms is mostly about predicting their computational time.
- Using the RAM model, we can count how many steps any algorithm takes on any given input instance.



Analyzing Running Time

- We can determine the exact running time, but the extra precision is usually not worth the effort of computing it.

Analyzing Running Time

- We can determine the exact running time, but the extra precision is usually not worth the effort of computing it.
- As n grows, the multiplicative constants and lower order terms are dominated by higher order terms.

Analyzing Running Time

- We can determine the exact running time, but the extra precision is usually not worth the effort of computing it.
- As n grows, the multiplicative constants and lower order terms are dominated by higher order terms.
- Asymptotic Analysis allows us to study how the running time increases as the input size increases without bounds ignoring levels of detail that do not impact our comparison of algorithms.

Growth Rates of Common Functions

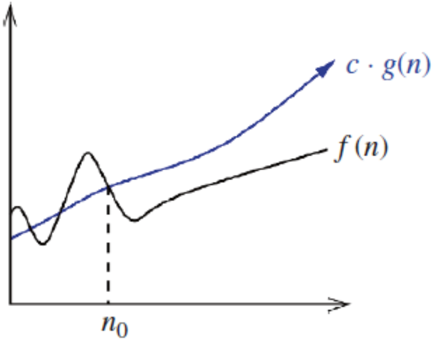
n	$\lg n$	n	$n \lg n$	n^2	2^n	$n!$
10	0.003 μs	0.01 μs	0.033 μs	0.1 μs	1 μs	3.63 ms
20	0.004 μs	0.02 μs	0.086 μs	0.4 μs	1 ms	77.1 years
30	0.005 μs	0.03 μs	0.147 μs	0.9 μs	1 sec	8.4×10^{15} yrs
40	0.005 μs	0.04 μs	0.213 μs	1.6 μs	18.3 min	
50	0.006 μs	0.05 μs	0.282 μs	2.5 μs	13 days	
100	0.007 μs	0.1 μs	0.644 μs	10 μs	4×10^{13} yrs	
1,000	0.010 μs	1.00 μs	9.966 μs	1 ms		
10,000	0.013 μs	10 μs	130 μs	100 ms		
100,000	0.017 μs	0.10 ms	1.67 ms	10 sec		
1,000,000	0.020 μs	1 ms	19.93 ms	16.7 min		
10,000,000	0.023 μs	0.01 sec	0.23 sec	1.16 days		
100,000,000	0.027 μs	0.10 sec	2.66 sec	115.7 days		
1,000,000,000	0.030 μs	1 sec	29.90 sec	31.7 years		

Figure: How long algorithms that use $f(n)$ operations take to run on a fast computer, where each operation costs one nanosecond.

Outline

1. Motivation
2. Asymptotic Notations
3. Recap

Big-Oh Notation - Asymptotic Upper Bound



Definition

$O(g(n)) = \{f(n) \mid$ there exist +ve constants c, n_0 such that $0 \leq f(n) \leq c\, g(n)$ for all $n \geq n_0$ $\}$.

Big-Oh Notation

GUC German University in Cairo

Key Takeaway

The statement $f(n) = O(g(n))$ means that the growth rate of $f(n)$ is no more than the growth rate of $g(n)$. It is useful for analyzing the worst case.

Big-Oh Notation

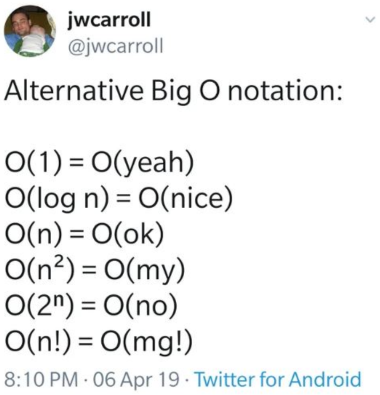
Key Takeaway

The statement $f(n) = O(g(n))$ means that the growth rate of $f(n)$ is no more than the growth rate of $g(n)$. It is useful for analyzing the worst case.

Big-Oh Rule

$f(n) = O(g(n))$ where $g(n)$ is $f(n)$ after dropping lower-order terms and constant factors.

Big-Oh Notation



jwcarroll
@jwcarroll

Alternative Big O notation:

$O(1) = O(\text{yeah})$
$O(\log n) = O(\text{nice})$
$O(n) = O(\text{ok})$
$O(n^2) = O(\text{my})$
$O(2^n) = O(\text{no})$
$O(n!) = O(\text{mg!})$

8:10 PM · 06 Apr 19 · Twitter for Android

Big-Oh Examples

GUC German University in Cairo

Example

- Is $2n + 10 = O(n)$?

Big-Oh Examples

Example

- Is $2n + 10 = O(n)$?

$$2n + 10 \leq cn$$
$$2 + \frac{10}{n} \leq c$$

Big-Oh Examples

Example

- Is $2n + 10 = O(n)$?

  $2n + 10 \leq cn$
  $2 + \frac{10}{n} \leq c$
  Choose $c = 3$ and $n_0 = 10$

- Is $3\ log(n) + 5 = O(log(n))$?

Big-Oh Examples

Example

- Is $2n + 10 = O(n)$?

  $2n + 10 \leq cn$
  $2 + \frac{10}{n} \leq c$
  Choose $c = 3$ and $n_0 = 10$

- Is $3\ log(n) + 5 = O(log(n))$?

  $3\ log(n) + 5 \leq c\ log(n)$
  $3 + \frac{5}{log(n)} \leq c$

Big-Oh Examples

GUC German University in Cairo

Example

- Is $2n + 10 = O(n)$?

  $2n + 10 \leq cn$
  $2 + \frac{10}{n} \leq c$
  Choose $c = 3$ and $n_0 = 10$

- Is $3\ log(n) + 5 = O(log(n))$?

  $3\ log(n) + 5 \leq c\ log(n)$
  $3 + \frac{5}{log(n)} \leq c$
  Choose $c = 8$ and $n_0 = 2$

Big-Oh Example

Example

- Is $n^2 = O(n)$?

Big-Oh Example

Example

- Is $n^2 = O(n)$?

$$n^2 \leq cn$$
$$n \leq c$$

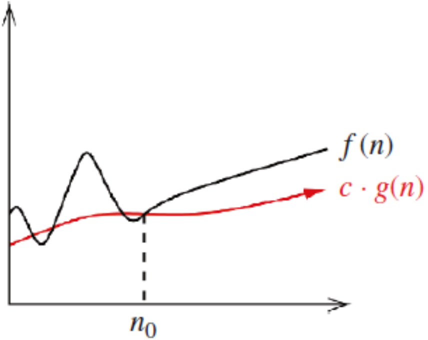
Big-Oh Example

Example

- Is $n^2 = O(n)$?

  $n^2 \leq cn$
  $n \leq c$
  Impossible to satisfy since c must be a constant!

Big-Omega Notation - Asymptotic Lower Bound



Definition

$\Omega(g(n)) = \{f(n) \mid$ there exist +ve constants c, n_0 such that $0 \leq c\ g(n) \leq f(n)$ for all $n \geq n_0$ $\}$.

Big-Omega Notation

GUC German University in Cairo

Key Takeaway

The statement $f(n) = \Omega(g(n))$ means that the growth rate of $f(n)$ is not less than the growth rate of $g(n)$. It is useful for analyzing the best case.

Big-Omega Examples

Example

- Is $2n^2 + 1 = \Omega(n)$?

Big-Omega Examples

GUC German University in Cairo

Example

- Is $2n^2 + 1 = \Omega(n)$?

$2n^2 + 1 \geq cn$
$2n + \frac{1}{n} \geq c$

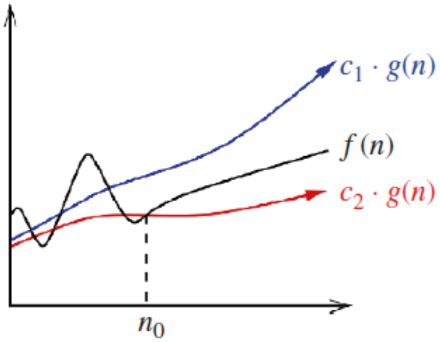
Big-Omega Examples

Example

- Is $2n^2 + 1 = \Omega(n)$?

  $2n^2 + 1 \geq cn$
  $2n + \frac{1}{n} \geq c$
  Choose $c = 3$ and $n_0 = 1$

Big-Theta Notation - Asymptotic Tight Bound



Definition

$\Theta(g(n)) = \{f(n) \mid$ there exist +ve constants c_1, c_2, n_0 such that $0 \leq c_2\ g(n) \leq f(n) \leq c_1\ g(n)$ for all $n \geq n_0$ $\}$.

Big-Theta Notation

GUC German University in Cairo

Key Takeaway

The statement $f(n) = \Theta(g(n))$ means that the growth rate of $f(n)$ is equal to the growth rate of $g(n)$.

Big-Theta Notation

GUC German University in Cairo

Key Takeaway

The statement $f(n) = \Theta(g(n))$ means that the growth rate of $f(n)$ is equal to the growth rate of $g(n)$.

Observation

If $f(n) = \Theta(g(n))$ then $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

Big-Theta Examples

Example

- Is $5n^2 = \Theta(n^2)$?

Big-Theta Examples

Example

- Is $5n^2 = \Theta(n^2)$?

  We need to show that $5n^2 = O(n^2)$ and $5n^2 = \Omega(n^2)$.

Big-Theta Examples

Example

- Is $5n^2 = \Theta(n^2)$?

  We need to show that $5n^2 = O(n^2)$ and $5n^2 = \Omega(n^2)$.
  $5n^2 \leq cn^2$
  $5 \leq c$

Big-Theta Examples

Example

- Is $5n^2 = \Theta(n^2)$?

  We need to show that $5n^2 = O(n^2)$ and $5n^2 = \Omega(n^2)$.
  $5n^2 \leq cn^2$
  $5 \leq c$
  Choose $c = 5$ and $n_0 = 1$
  $5n^2 \geq cn^2$
  $5 \geq c$

Big-Theta Examples

Example

- Is $5n^2 = \Theta(n^2)$?

  We need to show that $5n^2 = O(n^2)$ and $5n^2 = \Omega(n^2)$.
  $5n^2 \leq cn^2$
  $5 \leq c$

  Choose $c = 5$ and $n_0 = 1$

  $5n^2 \geq cn^2$
  $5 \geq c$

  Choose $c = 5$ and $n_0 = 1$

Properties of Asymptotic Notations

1. Transitivity.
    - If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $f(n) = O(h(n))$.
    - If $f(n) = \Omega(g(n))$ and $g(n) = \Omega(h(n))$, then $f(n) = \Omega(h(n))$.
    - If $f(n) = \Theta(g(n))$ and $g(n) = \theta(h(n))$, then $f(n) = \Theta(h(n))$.

Properties of Asymptotic Notations

1. Transitivity.
   - If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $f(n) = O(h(n))$.
   - If $f(n) = \Omega(g(n))$ and $g(n) = \Omega(h(n))$, then $f(n) = \Omega(h(n))$.
   - If $f(n) = \Theta(g(n))$ and $g(n) = \theta(h(n))$, then $f(n) = \Theta(h(n))$.
2. Reflexivity.
   - $f(n) = O(f(n))$.
   - $f(n) = \Omega(f(n))$.
   - $f(n) = \Theta(f(n))$.

Properties of Asymptotic Notations

GUC German University in Cairo

1. Transitivity.
   - If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $f(n) = O(h(n))$.
   - If $f(n) = \Omega(g(n))$ and $g(n) = \Omega(h(n))$, then $f(n) = \Omega(h(n))$.
   - If $f(n) = \Theta(g(n))$ and $g(n) = \theta(h(n))$, then $f(n) = \Theta(h(n))$.
2. Reflexivity.
   - $f(n) = O(f(n))$.
   - $f(n) = \Omega(f(n))$.
   - $f(n) = \Theta(f(n))$.
3. Symmetry.
   - $f(n) = \Theta(g(n))$ iff $g(n) = \Theta(f(n))$.

Properties of Asymptotic Notations

GUC German University in Cairo

1. Transitivity.
   - If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $f(n) = O(h(n))$.
   - If $f(n) = \Omega(g(n))$ and $g(n) = \Omega(h(n))$, then $f(n) = \Omega(h(n))$.
   - If $f(n) = \Theta(g(n))$ and $g(n) = \theta(h(n))$, then $f(n) = \Theta(h(n))$.
2. Reflexivity.
   - $f(n) = O(f(n))$.
   - $f(n) = \Omega(f(n))$.
   - $f(n) = \Theta(f(n))$.
3. Symmetry.
   - $f(n) = \Theta(g(n))$ iff $g(n) = \Theta(f(n))$.
4. Complementarity.
   - $f(n) = O(g(n))$ iff $g(n) = \Omega(f(n))$.

Other Asymptotic Notations

Definition (Little-Oh Notation)

$o(g(n)) = \{f(n) \mid$ for any +ve constant c, there is a +ve constant n_0 such that $0 \leq f(n) < c\, g(n)$ for all $n \geq n_0$ $\}$.

Definition (Little-Omega Notation)

$\omega(g(n)) = \{f(n) \mid$ for any +ve constant c, there is an n_0 such that $0 \leq f(n) > c\, g(n)$ for all $n \geq n_0$ $\}$.

Little-Oh and Little-Omega

Key Takeaway

- The statement $f(n) = o(g(n))$ means that the growth rate of $f(n)$ is less than the growth rate of $g(n)$.
- The statement $f(n) = \omega(g(n))$ means that the growth rate of $f(n)$ is more than the growth rate of $g(n)$.

Little-Oh and Little-Omega

Key Takeaway

- The statement $f(n) = o(g(n))$ means that the growth rate of $f(n)$ is less than the growth rate of $g(n)$.
- The statement $f(n) = \omega(g(n))$ means that the growth rate of $f(n)$ is more than the growth rate of $g(n)$.

Example

$2n^2 = O(n^2)$ and $2n^2 = \Omega(n^2)$ but $2n^2 \neq o(n^2)$ and $2n^2 \neq \omega(n^2)$.

Little-Oh and Little-Omega

GUC German University in Cairo

Key Takeaway

- The statement $f(n) = o(g(n))$ means that the growth rate of $f(n)$ is less than the growth rate of $g(n)$.
- The statement $f(n) = \omega(g(n))$ means that the growth rate of $f(n)$ is more than the growth rate of $g(n)$.

Example

$2n^2 = O(n^2)$ and $2n^2 = \Omega(n^2)$ but $2n^2 \neq o(n^2)$ and $2n^2 \neq \omega(n^2)$.

Observation

If $f(n) = o(g(n))$, then $f(n) = O(g(n))$. If $f(n) = \omega(g(n))$, then $f(n) = \Omega(g(n))$.

The Limit Test

Asymptotic Notations and the Limit Test

If $\lim_{n\to\infty} \frac{f(n)}{g(n)}$

The Limit Test

Asymptotic Notations and the Limit Test

If $\lim_{n\to\infty} \frac{f(n)}{g(n)}$

1. $= 0$, then $f(n) = o(g(n))$.
2. $= \infty$, then $f(n) = w(g(n))$.
3. $= c \in \mathbb{R}^+$, then $f(n) = \Theta(g(n))$.

The Limit Test

Asymptotic Notations and the Limit Test

If $\lim_{n\to\infty} \frac{f(n)}{g(n)}$

1. $= 0$, then $f(n) = o(g(n))$.
2. $= \infty$, then $f(n) = w(g(n))$.
3. $= c \in \mathbb{R}^+$, then $f(n) = \Theta(g(n))$.
4. $\neq \infty$, then $f(n) = O(g(n))$.
5. $\neq 0$, then $f(n) = \Omega(g(n))$.

The Limit Test - Examples

Example

Show using the limit test that $4n^3 + 3n^2 = O(n^4)$ and $4n^3 + 3n^2 = \Omega(n^2)$.

The Limit Test - Examples

Example

Show using the limit test that $4n^3 + 3n^2 = O(n^4)$ and $4n^3 + 3n^2 = \Omega(n^2)$.

$\lim_{n\to\infty} \frac{4n^3+3n^2}{n^4}$
$\lim_{n\to\infty} \frac{4}{n} + \frac{3}{n^2}$
$= 0$

$\lim_{n\to\infty} \frac{4n^3+3n^2}{n^2}$
$\lim_{n\to\infty} 4n + 3$
$= \infty$

The Limit Test - Examples

Example

Show using the limit test that $log_{10}(n) = \Theta(ln(n))$.

The Limit Test - Examples

Example

Show using the limit test that $log_{10}(n) = \Theta(ln(n))$.

$\lim_{n\to\infty} \frac{log_{10}(n)}{ln(n)} = \lim_{n\to\infty} \frac{\frac{1}{n\ \ ln(10)}}{\frac{1}{n}}$
$= \frac{1}{ln(10)} = 0.434 \in \mathbb{R}^+$.

Useful Logarithmic Identities

- $log(n) = log_2(n)$
- $ln(n) = log_e(n)$
- $log^k(n) = (log(n))^k$
- $log_c(ab) = log_c(a) + log_c(b)$
- $log_c(\frac{a}{b}) = log_c(a) - log_c(b)$
- $log_b(a^n) = nlog_b(a)$
- $a^{log_b(c)} = c^{log_b(a)}$
- $log_b(a) = \frac{log_c(a)}{log_c(b)}$
- $\frac{d}{dn}ln(n) = \frac{1}{n}$.
- $\frac{d}{dn}log_b(n) = \frac{1}{n\ ln(b)}$.

Final Exercises (1/2)

Example

Algorithm A is in $O(n\ log\ n)$ and algorithm B is in $\Omega(n^2)$. Your friend then claims that given the same input, A always takes less time to run than B. Is your friend right?

Final Exercises (1/2)

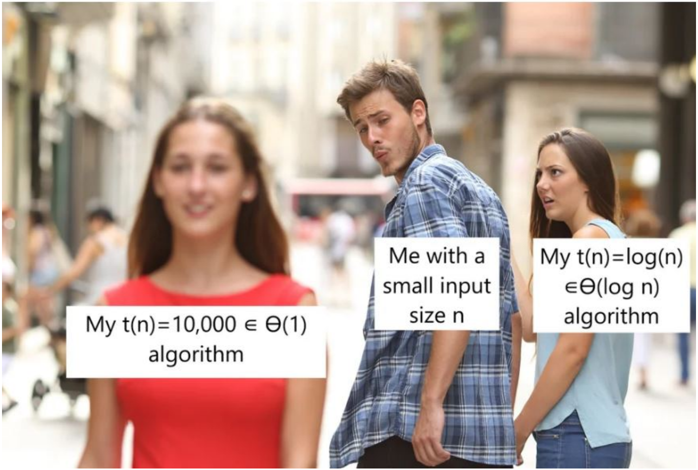
Example

Algorithm A is in $O(n\ log\ n)$ and algorithm B is in $\Omega(n^2)$. Your friend then claims that given the same input, A always takes less time to run than B. Is your friend right?

Key Takeaway

Larger asymptotic bound does not always mean faster!

Asympotic Analysis and Running Time



Final Exercises (2/2)

Example

A student has an algorithm which he proved (correctly) runs in $O(2^n)$. He coded the algorithm correctly, yet he was surprised when it ran quickly on inputs of size up to a million. What are possible explanations of this behavior?

Outline

Points to Take Home

1. Asymptotic notations.
2. The limit test.
3. Reading Material:
    - The Algorithm Design Manual, Chapter 2: Sections 2.1, 2.2, 2.3, and 2.4.
    - Introduction to Algorithms, Chapter 3: Section 3.1.

Next Lecture: Divide and Conquer I

Due Credits

The presented material is based on:

1. Previous editions of the course at the GUC due to Dr. Wael Aboulsaadat, Dr. Haythem Ismail, Dr. Amr Desouky, and Dr. Carmen Gervet.
2. Stony Brook University's Analysis of Algorithms Course.
3. MIT's Introduction to Algorithms Course.
4. Stanford's Design and Analysis of Algorithms Course.