CS483-03 Asymptotic Notations for Algorithm Analysis

Instructor: Fei Li

Room 443 ST II

Office hours: Tue. & Thur. 4:30pm - 5:30pm or by appointments

lifei@cs.gmu.edu with subject: CS483

http://www.cs.gmu.edu/∼ lifei/teaching/cs483_fall07/

Some of this lecture note is based on Dr. J.M. Lien's lecture notes.

Outline

- Review
- Order of growth
- O, Ω and Θ
- Examples and exercises

Review

In last class

- Analysis framework
  - **Input size**
  - **Basic operation**
  - **Running time**

$$T(n) \approx c_{op} \cdot C(n),$$

where

- n is the input size
- $C(n)$ is the total number of basic operations for input of size n.
- c_{op} is the time needed to execute one single basic operation.

  - **Worst-case, best-case, average-case**

Outline

- Review
- Order of growth
- O, Ω and Θ
- Examples and exercises

Orders of Growth

- **Running time** $T(n)$

$$T(n) \approx c_{op} \cdot C(n),$$

where n is the input size, $C(n)$ is the total number of basic operations for input of size n, and c_{op} is the time needed to execute one single basic operation.

- **Example** 1

Given that $C(n) = \frac{1}{2}n(n-1)$, how much the performance will be affected if the input size n is doubled?

$$growth = \frac{T(2n)}{T(n)} \approx \frac{c_{op} \cdot C(2n)}{c_{op} \cdot C(n)} = \frac{4n-2}{n-1} \approx 4$$

Orders of Growth

- **Running time** $T(n)$

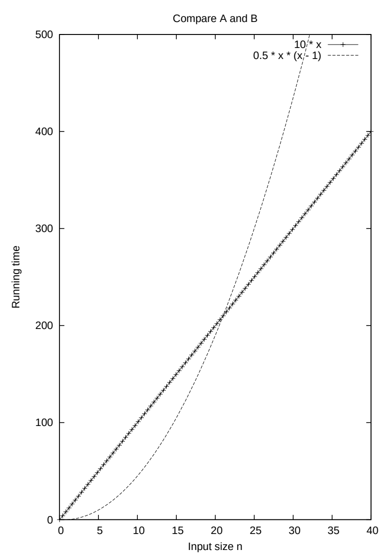
$$T(n) \approx c_{op} \cdot C(n),$$

where n is the input size, $C(n)$ is the total number of basic operations for input of size n, and c_{op} is the time needed to execute one single basic operation.

- **Example** 2

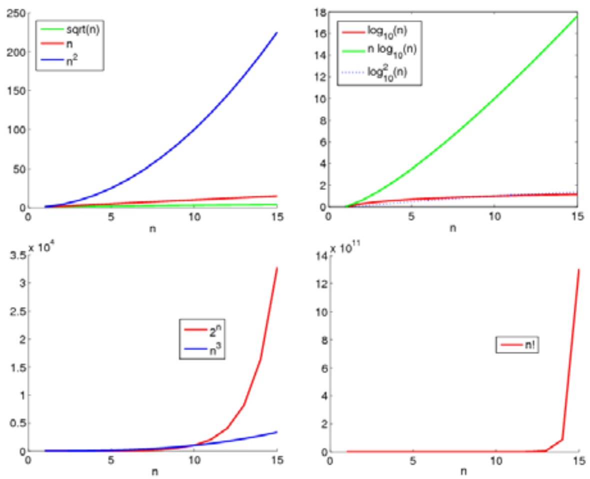
Given an algorithm A with $C_A(n) = 10 \cdot n$ and another algorithm B with $C_B(n) = \frac{1}{2}n(n-1)$, which algorithm is better?

$$\text{Answer} = \begin{cases} B, & \text{if } n \leq 21 \\ A, & \text{if } n > 21 \end{cases}$$



Orders of Growth

- Some of the commonly seen functions representing the number of the basic operations $C(n)$
  1. n (linear)
  2. n^2 (quadratic)
  3. n^3 (cubic)
  4. $\log_{10}(n)$ (logarithmic)
  5. $n \log_{10}(n)$ (n-log-n)
  6. $\log_{10}^2(n)$ (quadratic of $\log$)
  7. $\sqrt{n}$ (square root of n)
  8. 2^n (exponential)
  9. $n!$ (factorial function of n, 1808 by Christian Kramp)
- Can you order them by their growth rate?



Orders of Growth

n	n^2	n^3	2^n	$n!$
10	10^2	10^3	1024	3.6×10^6
100	10^4	10^6	1.3×10^{30}	9.3×10^{157}
1000	10^6	10^9	1.1×10^{301}	
10000	10^8	10^{12}		

n	$\log_{10}(n)$	$n \log_{10}(n)$	$\log^2_{10}(n)$	$\sqrt{n}$
10	1	10	1	3.16
100	2	200	4	10
1000	3	3000	9	31.6
10000	4	40000	16	100

Order the functions by their growth rate

$$\log_{10}(n) < \log^2_{10}(n) < \sqrt{n} < n < n \log_{10}(n) < n^2 < n^3 < 2^n < n!$$

Power of Growing Exponentially

from `http://www.ideum.com/portfolio`

- The king owns Shashi: $10,000,000,000,000,000,000$ grains of rice.

Outline

- Review
- Order of growth
- O, Ω and Θ
- Examples and exercises

Asymptotic Notations for Orders of Growth

- Ignore constant factors
- Ignore small input sizes
- Focus on order of growth
  - $O(g(n))$: a set of functions $f(n)$ that grow *no faster* than $g(n)$.
  - $\Omega(g(n))$: a set of functions $f(n)$ that grow *at least as fast* as $g(n)$.
  - $\Theta(g(n))$: a set of functions $f(n)$ that grow *at the same rate* as $g(n)$.

Asymptotic Notations

- $O(g(n)) := \{f(n) \mid$ there exist positive constants c and n_0 such that $0 \leq f(n) \leq c \cdot g(n)$ for all $n \geq n_0$ $\}$.

$$f(n) \in O(g(n))$$

$f(n)$ grow *no faster* than $g(n)$.

- $\Omega(g(n)) := \{f(n) \mid$ there exist positive constants c and n_0 such that $0 \leq c \cdot g(n) \leq f(n)$ for all $n \geq n_0$ $\}$.

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

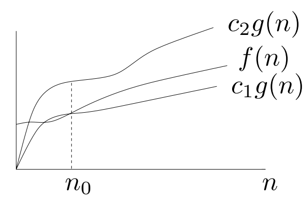
$f(n)$ grows *at least as fast* as $g(n)$.

- $\Theta(g(n)) := \{f(n) \mid$ there exist positive constants c_1, c_2, and n_0 such that $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ for all $n \geq n_0$ $\}$.

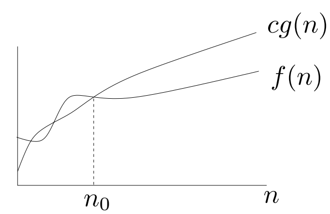
$$f(n) \in \Theta(g(n))$$

$f(n)$ grows *at the same rate* as $g(n)$.

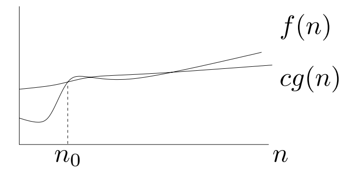
Asymptotic Notations



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



$f(n) \in O(g(n))$



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

Examples

- $O(g(n)) := \{f(n) \mid$ there exist positive constants c and n_0 such that $0 \leq f(n) \leq c \cdot g(n)$ for all $n \geq n_0$ $\}$.
  - $2n^2 - 5n + 1 \in O(n^2)$
  - $2^n + n^{100} - 2 \in O(n!)$
  - $2n + 6 \notin O(\log n)$
- $\Omega(g(n)) := \{f(n) \mid$ there exist positive constants c and n_0 such that $0 \leq c \cdot g(n) \leq f(n)$ for all $n \geq n_0$ $\}$.
- For any 2 functions $f(n)$ and $g(n)$, $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

Examples

- $O(g(n)) := \{f(n) \mid$ there exist positive constants c and n_0 such that $0 \leq f(n) \leq c \cdot g(n)$ for all $n \geq n_0$ $\}$.
  - $2n^2 - 5n + 1 \in O(n^2)$
  - $2^n + n^{100} - 2 \in O(n!)$
  - $2n + 6 \notin O(\log n)$
- $\Omega(g(n)) := \{f(n) \mid$ there exist positive constants c and n_0 such that $0 \leq c \cdot g(n) \leq f(n)$ for all $n \geq n_0$ $\}$.
  - $2n^2 - 5n + 1 \in \Omega(n^2)$
  - $2^n + n^{100} - 2 \notin \Omega(n!)$
  - $2n + 6 \in \Omega(\log n)$
- For any 2 functions $f(n)$ and $g(n)$, $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

Examples

- $O(g(n)) := \{f(n) \mid$ there exist positive constants c and n_0 such that $0 \leq f(n) \leq c \cdot g(n)$ for all $n \geq n_0$ $\}$.
  - $2n^2 - 5n + 1 \in O(n^2)$
  - $2^n + n^{100} - 2 \in O(n!)$
  - $2n + 6 \notin O(\log n)$
- $\Omega(g(n)) := \{f(n) \mid$ there exist positive constants c and n_0 such that $0 \leq c \cdot g(n) \leq f(n)$ for all $n \geq n_0$ $\}$.
  - $2n^2 - 5n + 1 \in \Omega(n^2)$
  - $2^n + n^{100} - 2 \notin \Omega(n!)$
  - $2n + 6 \in \Omega(\log n)$
- For any 2 functions $f(n)$ and $g(n)$, $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.
  - $2n^2 - 5n + 1 \in \Theta(n^2)$
  - $2^n + n^{100} - 2 \notin \Theta(n!)$
  - $2n + 6 \notin \Theta(\log n)$

Establish Order of Growth

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \begin{cases} 0 & f(n) \text{ has a smaller order of growth than } g(n) \\ c > 0 & f(n) \text{ has the same order of growth as } g(n) \\ \infty & f(n) \text{ has a larger order of growth than } g(n) \end{cases}$$

$$\begin{aligned} 2n^2 - 5n + 1 &\quad vs. \quad n^2 \\ 2^n + n^{100} - 2 &\quad vs. \quad n! \\ 2n + 6 &\quad vs. \quad \log n \end{aligned}$$

Establish Order of Growth

- **L'Hopital's rule**

  If $\lim_{n\to\infty} f(n) = \lim_{n\to\infty} g(n) = \infty$ and the derivatives f' and g' exist, then

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

- **Stirling's formula**

$$n! \approx \sqrt{2\pi n}(\frac{n}{e})^n$$

where e is the natural logarithm, $e \approx 2.718$. $\pi \approx 3.1415$.

$$\sqrt{2\pi n}(\frac{n}{e})^n \leq n! \leq \sqrt{2\pi n}(\frac{n}{e})^{n+\frac{1}{12n}}$$

Some Facts on Growth of Order

- A weak version of **Stirling's formula**

$$(\frac{n}{3})^n < n! < (\frac{n}{2})^n, \text{ if } n \geq 6.$$

- Another view

$$\ln n! \approx n \ln n - n + \frac{\ln n}{2} + \frac{\ln 2\pi}{2}$$

- $11!$ and $20!$ are the largest factorials stored in 32-bit and 64-bit computers. ($20! = 2432902008176640000$)
- googol is 10^{100} and $70! \approx 1.198$ googol.

  Consider arranging 70 people in a row.
- A googol is greater than the number of atoms in the observable universe, which has been variously estimated from 10^{79} up to 10^{81}.

Exercises

$$\begin{array}{rcl} 2n^2 - 5n + 1 & vs. & n^2 \\ 2^n + n^{100} - 2 & vs. & n! \\ 2n + 6 & vs. & \log n \end{array}$$

Exercises

- All logarithmic functions $\log_a n$ belong to the same class $\Theta(\log n)$ no matter what the logarithmic base $a > 1$ is.
- All polynomials of the same degree k belong to the same class $a_k n^k + a_{k-1} n^{k-1} + \cdots + a_1 n + a_0 \in \Theta(n^k)$.
- Exponential functions a^n have different orders of growth for different a's, i.e., $2^n \notin \Theta(3^n)$.
- $\Theta(\log n) < \Theta(n^a) < \Theta(a^n) < \Theta(n!) < \Theta(n^n)$, where $a > 1$.