Announcement

Reminder:

Homework 1: (Due on Jan 20, Thursday)

Last Class: Theoretical Analysis of Time Efficiency

Time efficiency is analyzed by determining the number of repetitions of the *<u>basic operation</u>* as a function of *<u>input size</u>*

<u>Basic operation:</u> **the operation that contributes most towards the running time of the algorithm.**

- ***n*** is a natural number (nonnegative integer)
- Sometimes, we can have multiple numbers.

input size

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

running time

execution time for basic operation

Number of times basic operation is executed

How to Choose Basic Operations

Basic operation should be simple and cannot be represented by other operations in the same algorithm

$$T(n) = T_1(n) + \cdots + T_M(n) = c_{op,1}t_1(n) + \cdots + c_{op,M}t_M(n)$$

If $\boldsymbol{t_1(n) \approx t_2(n), c_{op,1} \gg c_{op,2}}$ ➡ **Operation 1 is the basic operation**

If $\boldsymbol{t_1(n) \ll t_2(n), c_{op,1} \approx c_{op,2}}$ ➡ **Operation 2 is the basic operation**

Examples of Order of Growth

$c(n)$

n	$\log_2 n$	n	$n \log_2 n$	n^2	n^3	2^n	$n!$
10	3.3	10^1	$3.3 \cdot 10^1$	10^2	10^3	10^3	$3.6 \cdot 10^6$
10^2	6.6	10^2	$6.6 \cdot 10^2$	10^4	10^6	$1.3 \cdot 10^{30}$	$9.3 \cdot 10^{157}$
10^3	10	10^3	$1.0 \cdot 10^4$	10^6	10^9		
10^4	13	10^4	$1.3 \cdot 10^5$	10^8	10^{12}		
10^5	17	10^5	$1.7 \cdot 10^6$	10^{10}	10^{15}		
10^6	20	10^6	$2.0 \cdot 10^7$	10^{12}	10^{18}		

Time complexity increases!

Sample Run Time – Importance of Algorithm Design

n	ALPHA 21164A, C, Cubic Alg. (n^3)	TRS-80, BASIC, Linear Alg. (n)
10	0.6 microsecs	200 millisecs
100	0.6 millisecs	2.0 secs
1000	0.6 secs	20 secs
10,000	10 mins	3.2 mins
100,000	7 days	32 mins
1,000,000	19 yrs	5.4 hrs

Order of Growth

- **The number of basic operations grows with an increase of the input size**
- **Order of growth determines the growth rate as the input size goes to infinite**

Question: given two functions, how to compare the order of growth?

Asymptotic Growth Rate

A way of comparing functions that ignores constant factors and small input sizes:

- Compare the functions when ***n*** becomes sufficient large
- Constant factor will not affect asymptotic growth rate

Example:

Which function grows faster?

$$\frac{1}{4}n^2 \text{ or } 16n$$

Objective: given a simple function $g(n)$ with known running time, establish the relationship between a function $t(n)$ and $g(n)$ in terms of the order of growth

$$\text{e.g., } g(n) = 1, or\ n,\ or\ n^2, etc.$$

Asymptotic Growth Rate

$t(n)$ **is an algorithm's running time**

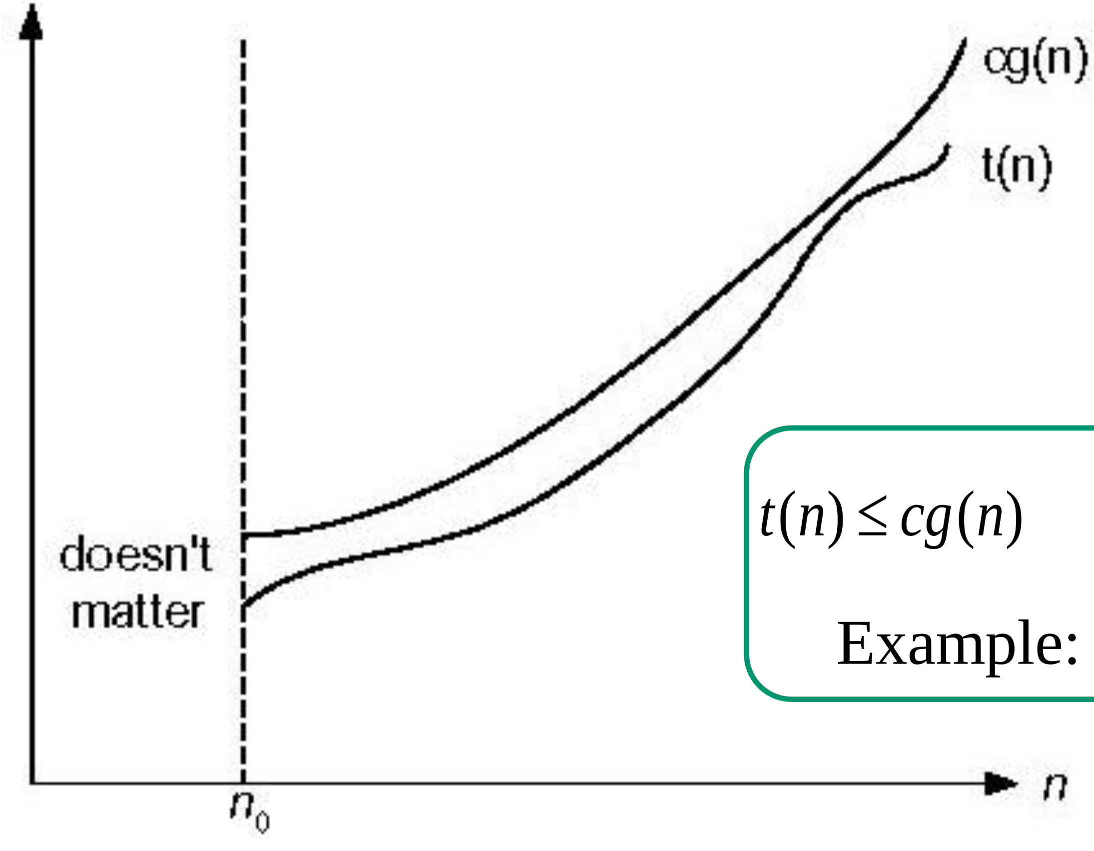
$g(n)$ **is a simple function of the running time**

$c_1 * t(n) + c_2$

- $\boldsymbol{O(g(n))}$: **class of functions $t(n)$ that grow *<u>no faster</u>* than $g(n)$**
  - –e.g., $500 \in O(n)$, $n + 100 \in O(n)$, and $500n \in O(n)$
- $\boldsymbol{\Theta(g(n))}$: **class of functions $t(n)$ that grow *<u>at same rate</u>* as $g(n)$**
  - –e.g., $n + 100 \in \Theta(n)$, and $500n \in \Theta(n)$
- $\boldsymbol{\Omega(g(n))}$: **class of functions $t(n)$ that grow *<u>at least as fast</u>* as $g(n)$**
  - –e.g., $0.001n^2 \in \Omega(n)$, $n + 100 \in \Omega(n)$, and $500n \in \Omega(n)$

Big-O - O(*g*(*n*))

O(*g*(*n*)): class of functions *t*(*n*) that grow *no faster* than *g*(*n*)



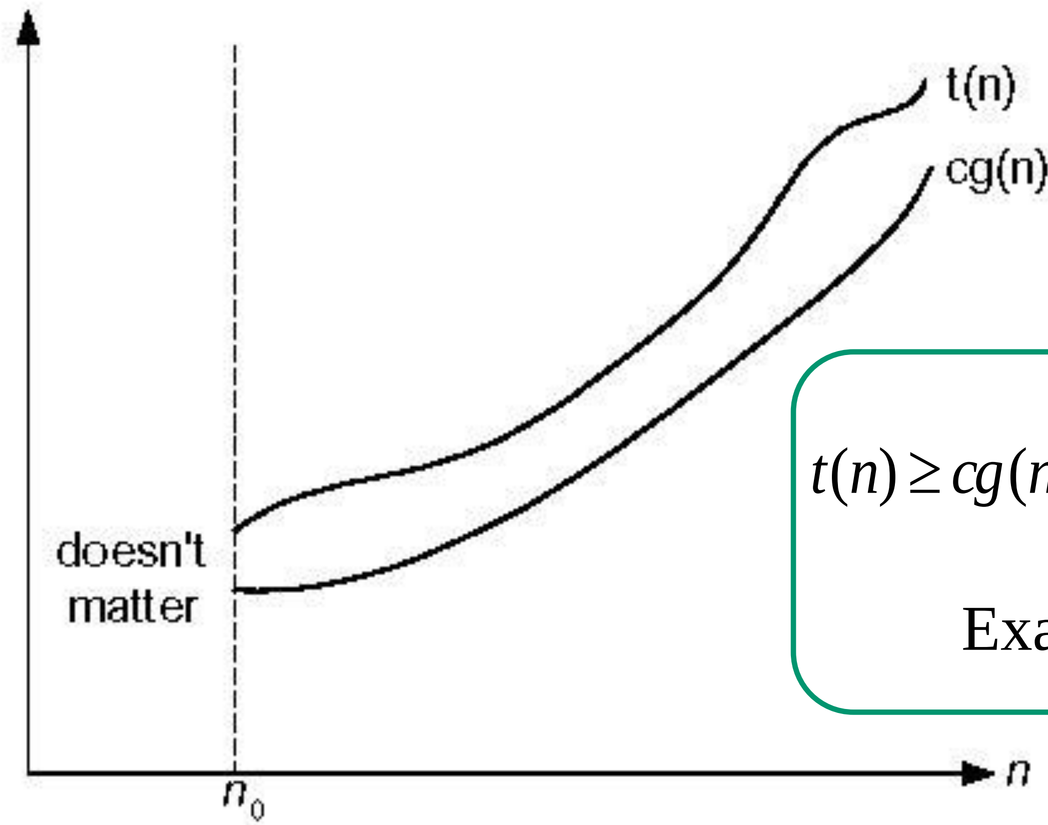
$$t(n) \in O(g(n))$$

$$t(n) \le cg(n) \quad for \quad \exists c > 0, \quad n \ge n_0, \quad n_0 \ge 0$$

Example: $2n \in O(n^2)$

Big-omega - $\Omega(g(n))$

$\Omega(g(n))$: class of functions $t(n)$ that grow *at least as fast* as $g(n)$



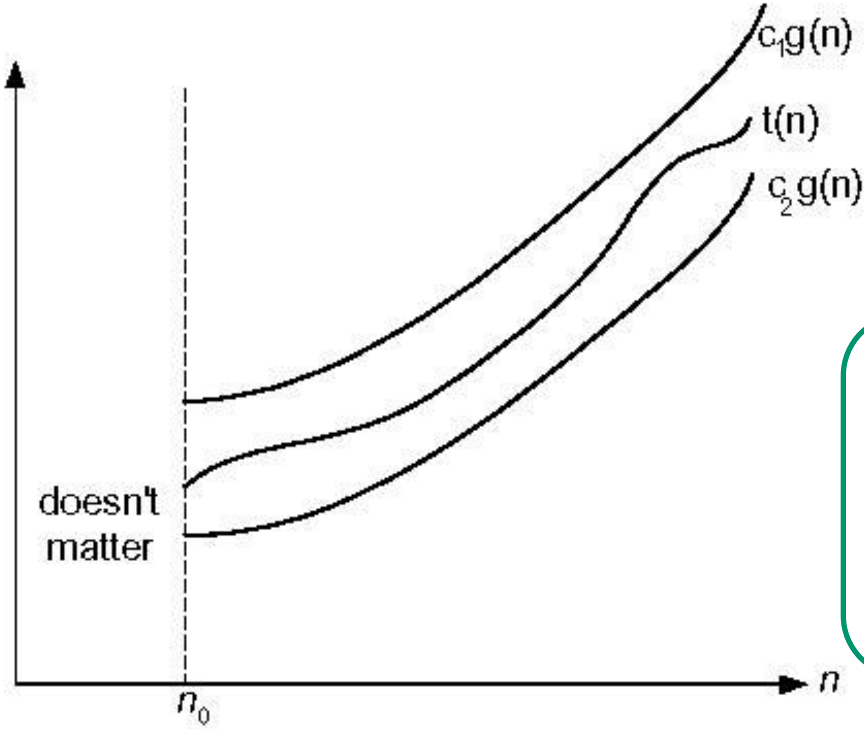
$$t(n) \in \Omega(g(n))$$

$$t(n) \geq cg(n) \quad for \quad \forall c > 0, \quad n \geq n_0, \quad n_0 \geq 0$$

Example: $n^2 \in \Omega(n)$

Big-theta - Θ (*g*(*n*))

Θ (*g*(*n*)): class of functions *t*(*n*) that grow *at same rate* as *g*(*n*)



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

$$c_1 g(n) \le t(n) \le c_2 g(n) \qquad n \ge n_0$$

Example:

$$n + 1 \in \Theta(n)$$

$$t(n) \in \Theta(g(n)) \Rightarrow t(n) \in O(g(n)) \text{ and } t(n) \in \Omega(g(n))$$

Relationships among Big-O, Big-omega, and Big-theta

1. If $t(n) \in O(g(n)),$ *then* $g(n) \in \Omega(t(n))$

E.g, $n \in O(n^2)$ and $n^2 \in \Omega(n)$

2. If $t(n) \in \Theta(g(n)),$

then $t(n) \in O(g(n))$ *and* $t(n) \in \Omega(g(n))$

E.g, $n + 1 \in \Theta(n)$, then $n + 1 \in O(n)$ and $n + 1 \in \Omega(n)$

Common Properties of Big O, Big-omega and Big-theta

1. If $t(n) \in O(g(n))$, *then* $ct(n) \in O(g(n))$

   Example: $cn \in O(n)$

2. If $t_1(n) \in O(g_1(n))$ and $t_2(n) \in O(g_2(n))$

   then $t_1(n)t_2(n) \in O(g_1(n)g_2(n))$

   Example: $n^2 * n \in O(n^5)$

3. If $t(n) \in O(g(n))$ and $g(n) \in O(h(n))$

   then $t(n) \in O(h(n))$

   Example: $n^2 \in O(n^5)$ and $n^5 \in O(n^6)$ ⇒ $n^2 \in O(n^6)$

Same property works for both Big-omega and Big-theta!

Properties of Big O and Big-Theta

If $t_1(n) \in O(g_1(n))$ and $t_2(n) \in O(g_2(n))$

then $t_1(n) + t_2(n) \in O(\mathbf{max}(g_1(n), g_2(n)))$

Example: $n^2 + n \in O(n^2)$

Same property works for Big-theta!

Properties of Big-omega

If $t_1(n) \in \Omega(g_1(n))$ and $t_2(n) \in \Omega(g_2(n))$

$$t_1(n) + t_2(n) \in \Omega(\mathbf{min}(g_1(n), g_2(n)))$$

Example: $n^2 + n \in \Omega(n)$

How to Establish Order of Growth

Based on the properties, we need to find an appropriate simple function $g(n)$ given a $t(n)$.

For example, can you find the simplest $g(n)$ for the functions below?

$$5n + 20 \in O(?)$$

$$0.5n + 100 \log n \in O(?)$$

$$2^n + n^2 \in O(?)$$

Establishing order of growth

Once you find $g(n)$

How can we prove our assertion $t(n) \in O(g(n))$?

- **Method 1: using definition**
- **Method 2: computing** $lim_{n \to \infty} t(n)/g(n)$

Prove the order of growth: Method 1 – using definition

$t(n) \in O(g(n))$ **if order of growth of *f*(*n*) ≤ order of growth of *g*(*n*) (within constant multiple)**

There exists a positive constant *c* and non-negative integer n_0 such that

$$t(n) \le cg(n) \quad n \ge n_0$$ ⟹ Find a constant c and n_0

Examples:

$$10n \in O(2n^2)$$

$$5n + 20 \in O(10n)$$

Prove the order of growth: Method 1 – using definition

More examples:

$4n^2 \in O(2n^2)$? Yes

$0.0000001n^2 \in O(n)$? No

$n \in O(n^2 - n)$? Yes

Prove the order of growth: Method 2 – using limits

$$\lim_{n\to\infty} T(n)/g(n) = \begin{cases} 0 & \text{order of growth of } T(n) \text{ \_\_\_ order of growth of } g(n) \\ c>0 & \text{order of growth of } T(n) \text{ \_\_\_ order of growth of } g(n) \\ \infty & \text{order of growth of } T(n) \text{ \_\_\_ order of growth of } g(n) \end{cases}$$

Prove the order of growth: Method 2 – using limits

$$\lim_{n\to\infty} T(n)/g(n) = \begin{cases} 0 & \text{order of growth of } T(n) \ \_\_\le\_ \ \text{order of growth of } g(n) \\ c>0 & \text{order of growth of } T(n) \ \_=\_\_ \ \text{order of growth of } g(n) \\ \infty & \text{order of growth of } T(n) \ \_\ge\_\_ \ \text{order of growth of } g(n) \end{cases}$$

$$t(n) \in O(g(n)) \quad \Omega(g(n)) \quad \Theta(g(n)) \quad ?$$

Prove the order of growth: Method 2 – using limits

$$\lim_{n\to\infty} T(n)/g(n) = \begin{cases} 0 & t(n) \in O(g(n)) \\ c>0 & t(n) \in \Theta(g(n)) \\ \infty & t(n) \in \Omega(g(n)) \end{cases}$$

Examples:

- $10n$ vs. $2n^2$
- $n(n+1)/2$ vs. n^2
- $\log_b n$ vs. $\log_c n$ $(b>c>1)$

L'Hôpital's Rule

If $\lim_{n\to\infty} f(n) = \lim_{n\to\infty} g(n) = \infty$ **and**

The derivatives $f'(n)$ **and** $g'(n)$ **exist,**

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

Example: $\dfrac{\log n}{\sqrt{n}}$

$\dfrac{n!}{2^n}$ $\quad n! \approx \sqrt{2\pi n}\left(\dfrac{n}{e}\right)^n$ (Stirling's formula)

Basic Asymptotic Efficiency Classes

Order of growth increase

1	constant
log n	logarithmic
n	linear
n log n	n log n
n^2	quadratic
n^3	cubic
2^n	exponential
$n!$	factorial

Examples

Compare the functions

$$n! \quad \text{vs.} \quad 2^n$$

$$3^n \quad \text{vs.} \quad 2^n$$

$$n \quad \text{vs.} \quad \ln^2 n$$

$$\log_2 n \quad \text{vs.} \quad \sqrt{n}$$

A useful formula: Stirling's formula

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

Some Important Properties of Order of Growth

- **All logarithmic functions $log_a n$ belong to the same class $\Theta(\log n)$ no matter what the logarithm's base $a > 1$ is**

- **All polynomials of the same degree k belong to the same class:**

$$a_k n^k + a_{k-1} n^{k-1} + \ldots + a_0 \in \Theta(n^k)$$

- **Exponential functions a^n have different orders of growth for different a's**

- **order $\log n$ < order n^α $(\alpha > 0)$ < order a^n < order $n!$ < order n^n**

Reading Assignments

Review how to calculate the derivative for simple functions from your Calculus classes

Chapter 2.2-2.3 and Appendix A

Announcement

We will have an in-class quiz (Quiz #1) on Thursday, Jan 20. It is closed-book, but open-notes.

You will be given a function $t(n)$. You should find a simple function $g(n)$ such that $t(n) \in O(g(n))$. You also need to prove your assertion either use the definition or the limit.

Please make sure you bring your laptop to class.

Please let me know if you prefer a hard copy of the quiz.