Australian National University

COMP3600/6466 Algorithms

Lecture 2

S2 2016

Dr. Hassan Hijazi

Prof. Weifa Liang

Algorithm Analysis

Why do we need it?

Problem

New Algorithm

Existing Algorithms

Time is a very expensive commodity

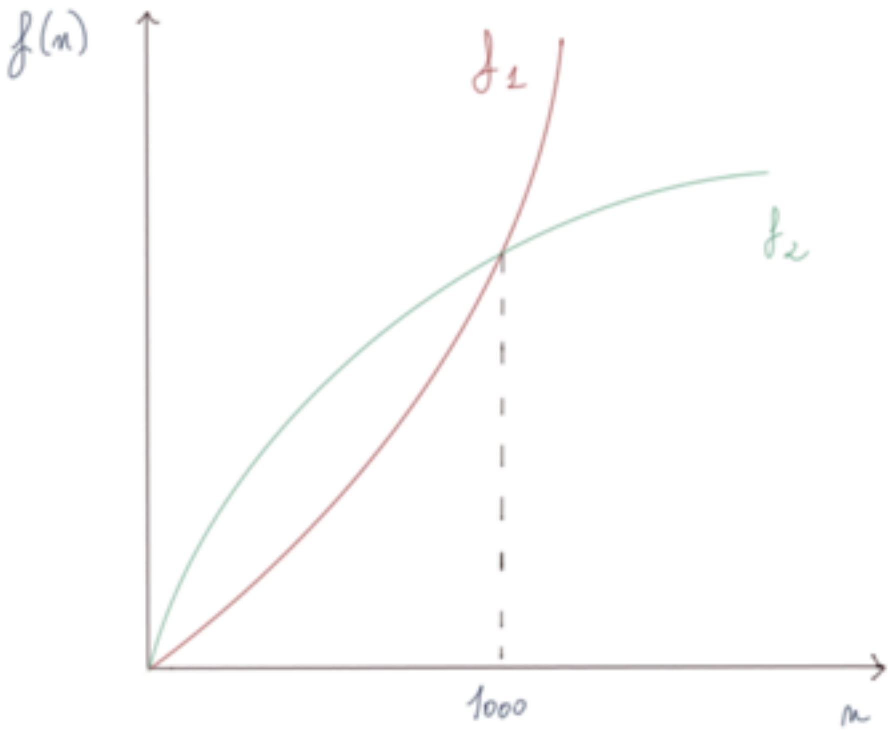
Algorithm Analysis

 VS 

Three Assumptions

1. Random Access Machine
2. Primitive Operations = Constant time
3. Asymptotic behavior

Asymptotic Behavior



Big-O Notation

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

$g(n)$ is an asymptotic upper bound for $f(n)$.

$f(n)$ grows at most as fast as $g(n)$.

Big-O Formal Definition

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

$$\Updownarrow$$

There **exist positive** c and n_0 such that $0 \leq f(n) \leq c \cdot g(n)$

for **all** $n \geq n_0$.

$$n^2 + n + 10 = O\left(n^2\right)?$$

Exercise 2.1

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

$$\Updownarrow$$

There **exist positive** c and n_0 such that $0 \leq f(n) \leq c \cdot g(n)$ for **all** $n \geq n_0$.

$$4n \lg n + 100n = O\left(n \lg n\right)?$$

Big-Omega Notation

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

$g(n)$ an asymptotic lower bound for $f(n)$.

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

Big-Omega Formal Definition

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

$$\Updownarrow$$

There **exist positive** c and n_0 such that $0 \leq c \cdot g(n) \leq f(n)$ for **all** $n \geq n_0$.

$$n^2 + n + 10 = \Omega\left(n^2\right)?$$

Exercise 2.2

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

$$\Updownarrow$$

There **exist positive** c and n_0 such that $0 \leq c \cdot g(n) \leq f(n)$ for **all** $n \geq n_0$.

$$4n \lg n + 100n = \Omega\left(n \lg n\right)?$$

Big-Theta Notation

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

$g(n)$ an asymptotically tight bound for $f(n)$.

$f(n)$ and $g(n)$ have the same rate of growth.

Big-Theta Formal Definition

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

$$\Updownarrow$$

There **exist** positive $\boldsymbol{c}1$, $\boldsymbol{c}2$ and $\boldsymbol{n}_0$ such that

$\boldsymbol{c}1 \cdot g(n) \leq f(n) \leq \boldsymbol{c}2 \cdot g(n)$, for **all** $n \geq \boldsymbol{n}_0$.

$$n^2 + n + 10 = O\left(n^2\right) \text{ and } n^2 + n + 10 = \Omega\left(n^2\right)$$

$$\Updownarrow$$

$$n^2 + n + 10 = \Theta\left(n^2\right)$$

Complexity of Algorithms

Assumption 3. Ignore constants and low order terms

$$2n^3 + 20n + 16 \equiv n^3$$

$$n^3 + 2n^2 + 5n + 10 \equiv n^3$$

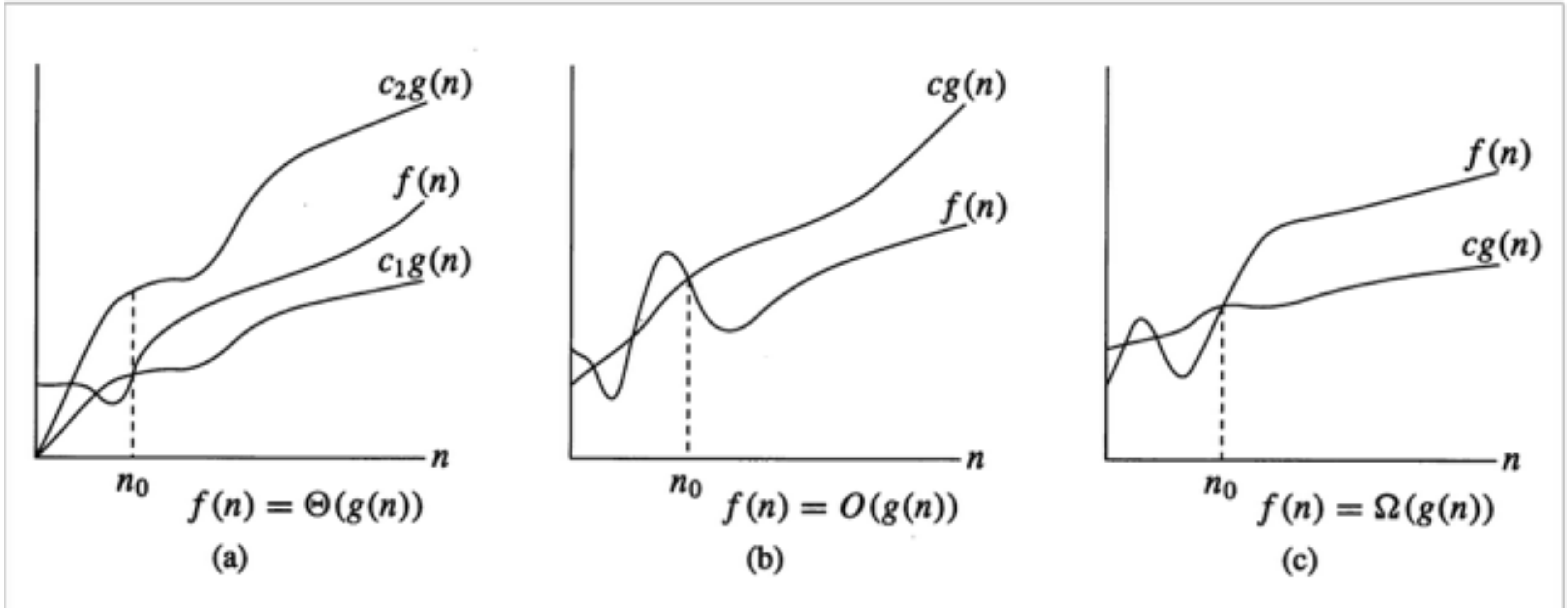
$$100n^2 + n \equiv n^2$$

Use the big-Theta notation

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

$g(n)$ an asymptotically tight bound for $f(n)$.

Asymptotic Behavior



Asymptotic Behavior

Some history about these asymptotic notations:

- Big-O notation introduced in 1892
- For 80 years, people used the big-O notation for both upper and lower bounds
- In 1976 Knuth advocates for the use of big-Omega and big-Theta notations to correct the popular, but technically sloppy, practice in the literature.
- Today, people still use the big-O notation in a sloppy fashion..

Not Asymptotically Tight..

In 1909, E. Landau needed to express the following idea:

This function is an asymptotic upper bound that is not tight

$2n^2 = O(n^2)$ is asymptotically tight, but $2n = O(n^2)$ is not.

o-notation $\equiv$ an upper bound that is not asymptotically tight.

Little-oh Notation

$2n^2 \neq o(n^2)$ $2n = o(n^2)$

$2n^2 = O(n^2)$ is asymptotically tight, but $2n = O(n^2)$ is not.

o-notation $\equiv$ an upper bound that is not asymptotically tight.

$$f(n) = o(g(n))$$

$$\Updownarrow$$

For **any** positive $\boldsymbol{c}$, there exists a positive $\boldsymbol{n}_0$ such that

$$0 \leq f(n) < \boldsymbol{c} \cdot g(n), \text{ for } \mathbf{all}\ n \geq \boldsymbol{n}_0.$$

Little-oh Notation

$$f(n) = o(g(n))$$

$$\Updownarrow$$

For **any** positive $\boldsymbol{c}$, there exists a positive $\boldsymbol{n}_0$ such that $0 \leq f(n) < \boldsymbol{c} \cdot g(n)$, for **all** $n \geq \boldsymbol{n}_0$.

If $f(n)$ and $g(n)$ are positive, this is equivalent to:

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

Little-oh Notation

$$f(n) = o(g(n))$$

$$\Updownarrow$$

For **any** positive $\boldsymbol{c}$, there exists a positive $\boldsymbol{n}_0$ such that

$0 \leq f(n) < \boldsymbol{c} \cdot g(n)$, for **all** $n \geq \boldsymbol{n}_0$.

$$2n = o\left(n^2\right)$$

$$\Updownarrow$$

$$\forall c > 0,\ \exists n_0 \text{ s.t. } 0 \leq 2n < c \cdot n^2, \forall n \geq n_0$$

$$\Updownarrow$$

$$\forall c > 0,\ \exists n_0 \text{ s.t. } 0 \leq \frac{2}{n} < c, \forall n \geq n_0$$

$$\Updownarrow$$

$$\forall c > 0,\ \text{Let } n_0 = \left\lceil \frac{3}{c} \right\rceil, \text{ then } 0 \leq \frac{2}{n} < c, \forall n \geq n_0$$

Exercise 2.3

$$2n^2 \neq o(n^2)$$

Little-omega Notation

$n^2 + n \neq \omega(n^2)$ $n^2 + n = \omega(n)$

$n^2 + n = \Omega(n^2)$ is asymptotically tight, but $n^2 + n = \Omega(n)$ is not.

ω-notation $\equiv$ a lower bound that is not asymptotically tight.

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

$$\Updownarrow$$

For **all** positive $\boldsymbol{c}$, there exists $\boldsymbol{n}_0$ such that

$0 \leq \boldsymbol{c} \cdot g(n) < f(n)$, for **all** $n \geq \boldsymbol{n}_0$.

Little-omega Notation

$$f(n) = \omega(g(n))$$
$$\Updownarrow$$

For **all** positive $\boldsymbol{c}$, there exists $\boldsymbol{n}_0$ such that

$$0 \leq \boldsymbol{c} \cdot g(n) < f(n), \text{ for } \textbf{all } n \geq \boldsymbol{n}_0.$$

If $f(n)$ and $g(n)$ are positive, this is equivalent to:

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

Little-omega Notation

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

$$\Updownarrow$$

For **all** positive $\boldsymbol{c}$, there exists $\boldsymbol{n}_0$ such that

$0 \leq \boldsymbol{c} \cdot g(n) < f(n)$, for **all** $n \geq \boldsymbol{n}_0$.

$$n^2 + n = \omega(n)$$

$$\Updownarrow$$

$$\forall c > 0,\ \exists n_0 \text{ s.t. } 0 \leq c \cdot n < n^2 + n, \forall n \geq n_0$$

$$\Updownarrow$$

$$\forall c > 0,\ \exists n_0 \text{ s.t. } 0 \leq c < n + 1, \forall n \geq n_0$$

$$\Updownarrow$$

$$\forall c > 0, \text{ let } n_0 = \lceil c \rceil, \text{ then } 0 \leq c < n + 1, \forall n \geq n_0,$$

Exercise 2.4

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

$$\Updownarrow$$

For **all** positive $\boldsymbol{c}$, there exists $\boldsymbol{n}_0$ such that

$0 \leq \boldsymbol{c} \cdot g(n) < f(n)$, for **all** $n \geq \boldsymbol{n}_0$.

$$n^2 + n \neq \omega(n^2)$$

Exercise 2.5

Show that for any real constants a and b, where $b > 0$,

$$(n + a)^b = \Theta(n^b)$$

Exercise 2.6

For each statement, select between

always true, never true, or sometimes true:

1. $f(n) = O(f(n)^2)$
2. $f(n) = \Omega(g(n))$ and $f(n) = o(g(n))$