Lecture 3.
Growth of Functions – Asymptotic Analysis

Introduction to Algorithms

Da Nang University of Science and Technology

Dang Thien Binh

dtbinh@dut.udn.vn

Overview of Asymptotic Notation

- A way to describe behavior of functions **in the limit**
  - We are studying asymptotic efficiency
- Describe growth of functions
- Focus on what is important by abstracting away low-order terms and constant factors
- How we indicate running times of algorithms
- A way to compare "sizes" of functions
  - $$\begin{array}{ccc} O & \approx & \leq \\ \Omega & \approx & \geq \\ \Theta & \approx & = \\ o & \approx & < \\ \omega & \approx & > \end{array}$$

Asymptotic Notation

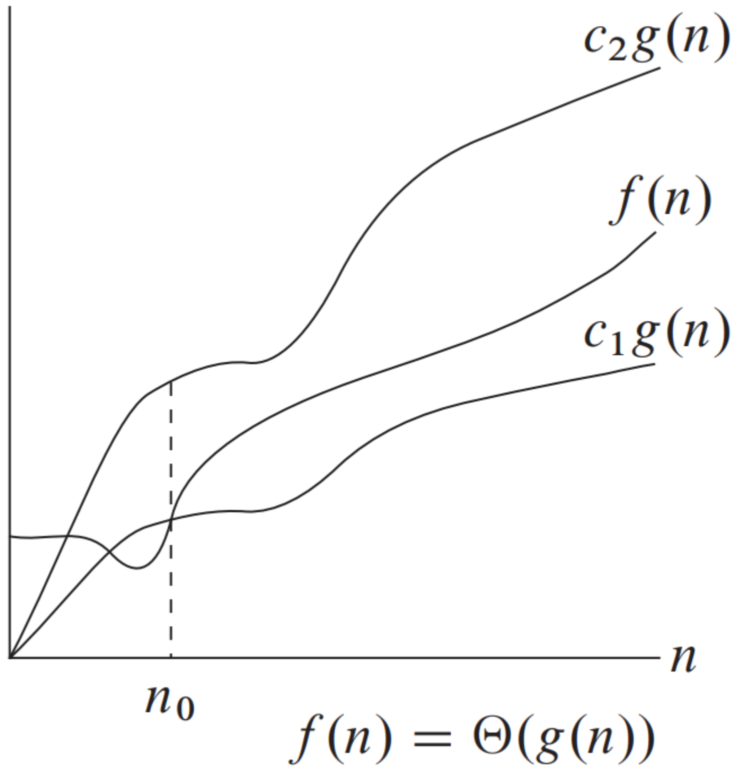
- ***Asymptotic efficiency of algorithms***
  - Concerned with how the running time increases with the size of the input in the limit
    - i.e., as the size of the input increases without bound.
  - An asymptotically more efficient algorithm is the best choice
- Asymptotic running time of an algorithm is defined in terms of functions whose domains are the set of natural numbers
  $\mathbf{N} = \{0, 1, 2, \ldots\}$

Θ - Notation (1/5)

- ***Θ - notation (theta)***
  - $\Theta(g(n))$ = {$f(n)$: there exist positive constants c_1, c_2, and n_0 such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0$}
  - $\Theta(g(n))$ is the set of functions
  - $f(n) = \Theta(g(n))$ really means that $f(n)$ belongs to $\Theta(g(n))$
  - *g(n)* is called an ***asymptotically tight bound*** for *f(n)*

Θ - Notation (2/5)

- ***Θ - notation***



$g(n)$ is an **asymptotically tight bound** for $f(n)$

Θ - Notation (3/5)

- ***Θ - notation***
  - If $f(n)$ is a polynomial of degree d, then $f(n) = \Theta(n^d)$
  - *Example*:
    - $n^2 - 2n = \Theta(n^2)$
    - $200n^2 - 100n = \Theta(n^2)$

$$c_1 n^2 \le n^2 - 2n \le c_2 n^2 \qquad c_1 n^2 \le 200n^2 - 100n \le c_2 n^2$$

$$c_1 \le 1 - \frac{2}{n} \le c_2 \qquad c_1 \le 200 - \frac{100}{n} \le c_2$$

$$c_1 \le \frac{1}{3} \quad c_2 \ge 1 \qquad c_1 \le 100 \quad c_2 \ge 200$$

$$n \ge 3 \quad n \ge 1 \qquad n \ge 1 \quad n \ge 1$$

$$n_0 = 3 \qquad n_0 = 1$$

    - Some choice for c_1, c_2, and n_0 exists, the functions are $\Theta(n^2)$
    - $\Theta(n^0)$?

Θ - Notation (4/5)

- $n^2 - 2n = \Theta(n^2)$

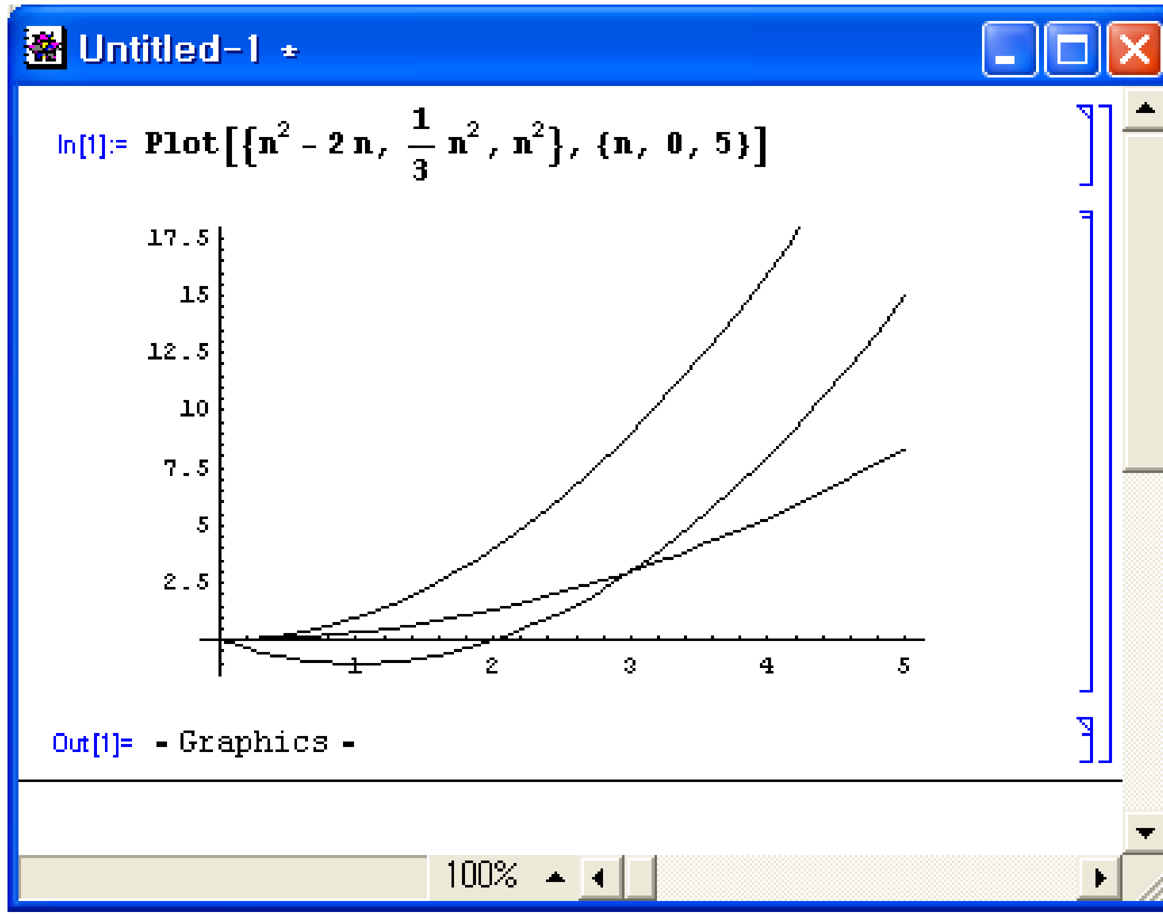
$$c_1 n^2 \le n^2 - 2n \le c_2 n^2$$

$$c_1 \le 1 - \frac{2}{n} \le c_2$$

$$c_1 \le \frac{1}{3} \qquad c_2 \ge 1$$

$$n \ge 3 \qquad n \ge 1$$

$$n_0 = 3$$



Θ - Notation (5/5)

- $200n^2 - 100n = \Theta(n^2)$

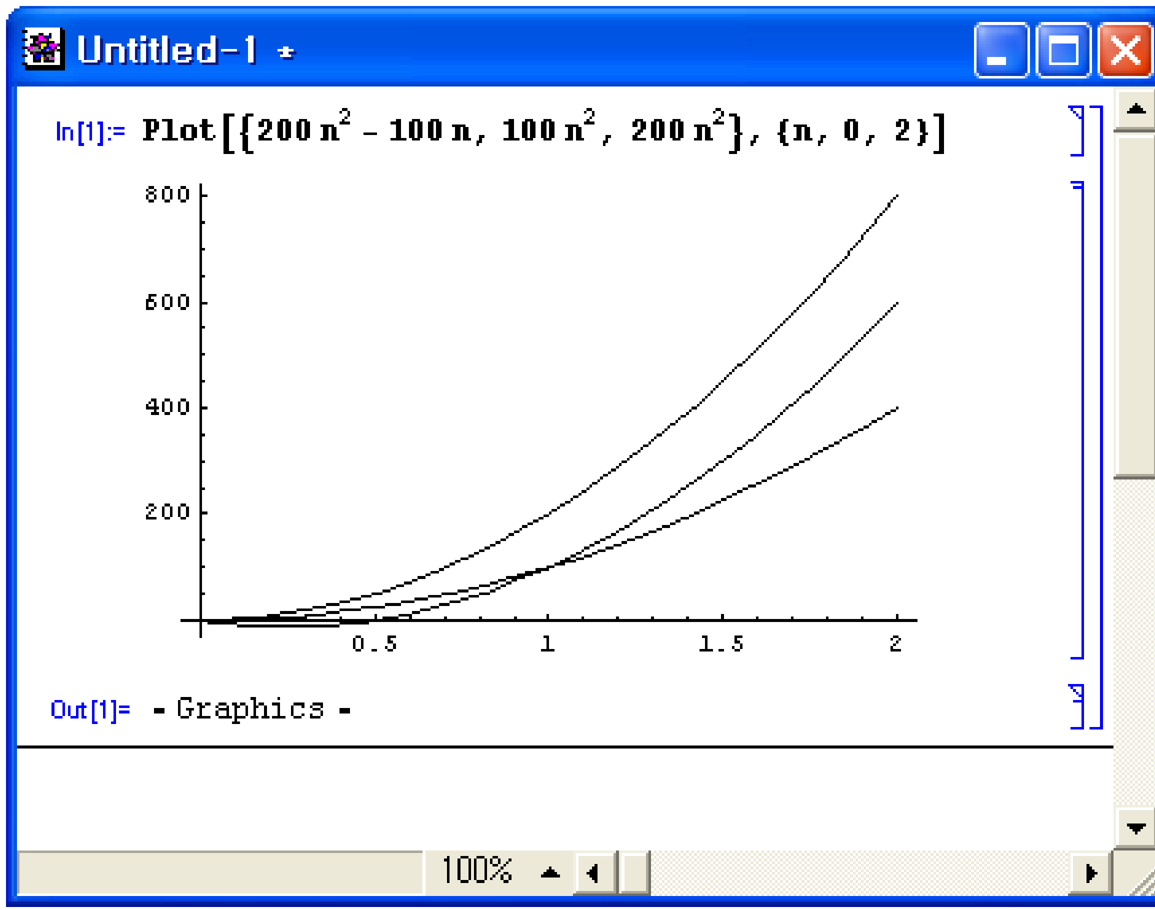
$$c_1 n^2 \le 200n^2 - 100n \le c_2 n^2$$

$$c_1 \le 200 - \frac{100}{n} \le c_2$$

$$c_1 \le 100 \qquad c_2 \ge 200$$

$$n \ge 1 \qquad n \ge 1$$

$$n_0 = 1$$

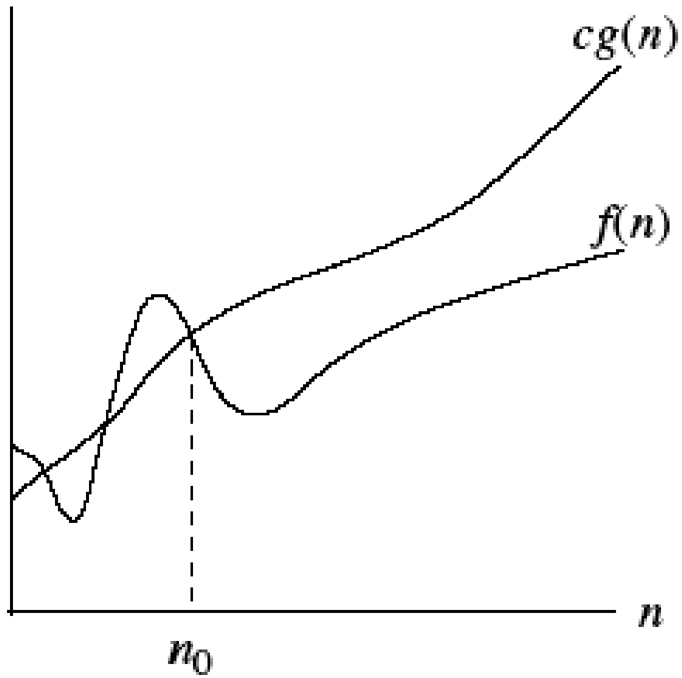


O – Notation (1/2)

- ***O* - notation (big-oh)**
  - $f(n) = \boldsymbol{O}(g(n))$: $g(n)$ is an *asymptotic upper bound* for $f(n)$
  - $\boldsymbol{O}(g(n)) = \{f(n)$: there exist positive constants c and n_0 such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0\}$
- $f(n) = \Theta(g(n))$ implies $f(n) = \boldsymbol{O}(g(n))$, but $f(n) = \boldsymbol{O}(g(n))$ does NOT imply $f(n) = \Theta(g(n))$
- ***Example***:
  - $n^2 - 2n = \boldsymbol{O}(n^2)$
  - $200n^2 - 100n = \boldsymbol{O}(n^2) = \boldsymbol{O}(n^3) = \ldots$
  - $n = \boldsymbol{O}(n^2)$
- $\boldsymbol{O}$ is good for describing the *worst-case running time* of an algorithm

O – Notation (2/2)

- ***O - notation***



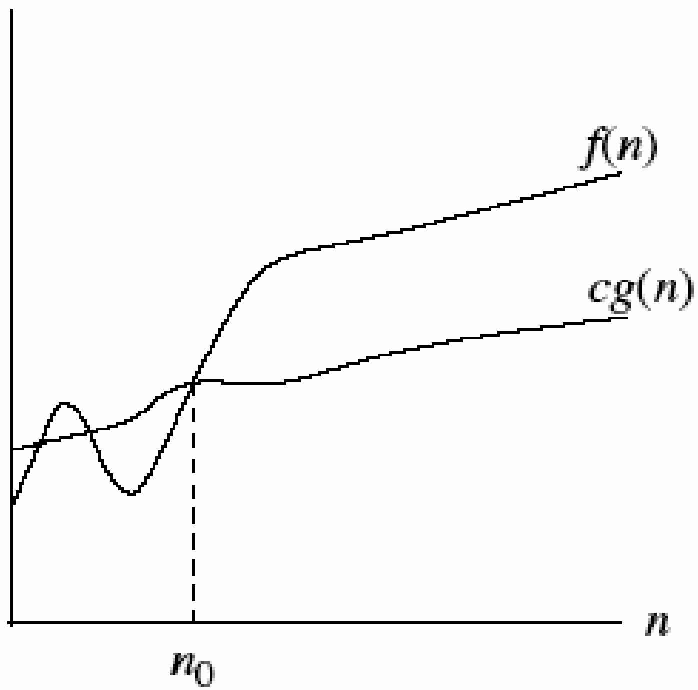
$g(n)$ is an ***asymptotic upper bound*** for $f(n)$.

Ω - Notation (1/2)

- ***Ω - notation (omega)***
  - $f(n) = \Omega(g(n))$: $g(n)$ is an *asymptotic lower bound* for $f(n)$
  - $\Omega(g(n)) = \{f(n)$: there exist positive constants c and n_0 such that 0 $0 \le cg(n) \le f(n)$ for all $n \ge n_0$ $\}$
- $f(n) = \Theta(g(n))$ implies $f(n) = \Omega(g(n))$,
  but $f(n) = \Omega(g(n))$ does NOT imply $f(n) = \Theta(g(n))$
- ***Example***:
  - $n^2 - 2n = \Omega(n^2)$
  - $200n^2 - 100n = \Omega(n^2) = \Omega(n) = \Omega(1)$
  - $n^2 = \Omega(n)$
- Ω is good for describing the *best case running time* of an algorithm
- ***Theorem***
  - $f(n) = \Theta(g(n))$ if and only if $f(n) = \boldsymbol{O}(g(n))$ and $f(n) = \Omega(g(n))$

Ω - Notation (2/2)

- ***Ω - notation (omega)***



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

Asymptotic Notation in Equations

- On the right-hand-side
  - $n = \boldsymbol{O}(n^2)$ means "n belongs to $\boldsymbol{O}(n^2)$"
- In general, asymptotic notation stands for some anonymous function
- ***Example***:
  - $2n^2 + 3n + 1 = 2n^2 + \Theta(n)$ means
    - there is some function $f(n) \in \Theta(n)$ that makes the equation true, i.e. $f(n) = 3n + 1$
  - $2n^2 + \Theta(n) = \Theta(n^2)$ means
    - for any function $f(n) \in \Theta(n)$, there is some function $g(n) \in \Theta(n^2)$ such that $2n^2 + f(n) = g(n)$ for all n
  - $2n^2 + 3n + 1 = 2n^2 + \Theta(n) = \Theta(n^2)$ (a chain of relationships)

o - Notation

- ***o* - notation (little-oh)**
  - ***O* - notation** may or may not be asymptotically tight
  - $2n^2 = O(n^2)$ is asymptotically tight, but $2n = O(n^2)$ is not
  - $f(n) = o(g(n))$: $g(n)$ is an *upper bound* of $f(n)$ that is *not asymptotically tight*
  - $o(g(n)) = \{f(n)$: for any positive constant $c > 0$, there exists a constant $n_0 > 0$ such that $0 \le f(n) < cg(n)$ for all $n \ge n_0\}$
  - $2n = o(n^2)$, but $2n^2 \neq o(n^2)$
- ***O***: for some constant *c*, ***o***: for all constant *c*
- In the ***o-notation***, the function *f(n)* becomes *insignificant* relative to *g(n)* as *n* approaches infinity
  - i.e., $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$

ω - Notation

- ***ω - notation (little-omega)***
  - $2n^2 = \Omega(n^2)$ is asymptotically tight, but $2n^3 = \Omega(n^2)$ is not
  - *f(n)* = ***ω*** (*g(n)*): *g(n)* is a *<u>lower bound</u>* of *f(n)* that is *<u>not asymptotically tight</u>*
  - $\omega(g(n))$ = {*f(n)*: for any positive constant *c* > 0, there exists a constant $n_0 > 0$ such that $0 \leq cg(n) < f(n)$ for all $n \geq n_0$}
  - $2n^2 = \omega(n)$, but $2n^2 \neq \omega(n^2)$
- In the ***ω-notation***, the function *f(n)* becomes *<u>arbitrarily large</u>* relative to *g(n)* as *n* approaches infinity
  - i.e., $\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty$

Comparison of Functions (1/2)

- *Many of the relational properties of real numbers apply to asymptotic comparisons too*
  - ***Reflexivity***
    - $f(n) = \Theta(f(n))$
    - *This is true for* $\mathbf{O}$, Ω
  - ***Symmetry***
    - $f(n) = \Theta(g(n))$ *if and only if* $g(n) = \Theta(f(n))$
  - ***Transpose symmetry***
    - $f(n) = \mathbf{O}(g(n))$ *if and only if* $g(n) = \Omega(f(n))$
    - $f(n) = \mathbf{o}(g(n))$ *if and only if* $g(n) = \omega(f(n))$
  - ***Transitivity***
    - $f(n) = \Theta(g(n))$ *and* $g(n) = \Theta(h(n))$ *imply* $f(n) = \Theta(h(n))$
    - *This is true for* $\mathbf{O}$, Ω, $\mathbf{o}$, *and* ω

Comparison of Functions (2/2)

- ***Analogy to real numbers***
  - $f(n) = \mathbf{O}(g(n)) \quad \approx \quad a \leq b$
  - $f(n) = \Omega(g(n)) \quad \approx \quad a \geq b$
  - $f(n) = \Theta(g(n)) \quad \approx \quad a = b$
  - $f(n) = \mathbf{o}(g(n)) \quad \approx \quad a < b$
  - $f(n) = \omega(g(n)) \quad \approx \quad a > b$
- ***Example***: $f(n) = 3n^3 + 4$
  - $f(n) = \Theta(n^3)$
  - $f(n) = \mathbf{O}(n^3) = \mathbf{O}(n^4) = \ldots$
  - $f(n) = \Omega(n^3) = \Omega(n^2) = \Omega(n) = \Omega(1)$
  - $f(n) = \mathbf{o}(n^4) = \mathbf{o}(n^5) = \ldots$
  - $f(n) = \omega(n^2) = \omega(n) = \omega(1)$

Standard Notation and Common Functions (1/2)

- ***Monotonicity***: A function *f(n)* is:
  - monotonically increasing if $a \leq b$ implies $f(a) \leq f(b)$
  - monotonically decreasing if $a \leq b$ implies $f(a) \geq f(b)$
  - strictly increasing if *a* < *b* implies *f(a)* < *f(b)*
  - strictly decreasing if *a* < *b* implies *f(a)* > *f(b)*
- ***Floors and ceilings***
  - Floor: $\lfloor x \rfloor$ is the greatest integer $\leq x$
  - Ceiling: $\lceil x \rceil$ is the least integer $\geq x$
  - ***Examples***:
    $\lfloor 3 \rfloor = 3$ $\lceil 3 \rceil = 3$
    $\lfloor 3.3 \rfloor = 3$ $\lceil 3.3 \rceil = 4$
    $\lfloor 3.9 \rfloor = 3$ $\lceil 3.9 \rceil = 4$
    $\lceil n/2 \rceil + \lfloor n/2 \rfloor = n, \quad \lceil \lceil n/a \rceil / b \rceil = \lceil n/ab \rceil, \quad \lfloor \lfloor n/a \rfloor / b \rfloor = \lfloor n/ab \rfloor$

Standard Notation and Common Functions (2/2)

- ***Logarithms***
  - $log_b\, n$: logarithm of *n* base *b*
  - $\lg n \ = \ \log_2 n$ (binary logarithm)
  - $\ln n \ = \ loge\, n$ (natural logarithm, e = 2.7182...)
- ***Factorials***
  - $n! = \begin{cases} 1 & if\ n = 0 \\ n(n-1)! & if\ n > 0 \end{cases}$
  - $n! = 1 * 2 * 3 * \ldots * n$
  - $n! \leq n^n$, thus $\boldsymbol{O}(n^n)$
- Stirling's Approximation
  - $n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1 + \Theta\left(\frac{1}{n}\right)\right)$
  - From Stirling's approximation, the followings hold:
    - $n! \ = \ \boldsymbol{o}(n^n)$
    - $n! \ = \ \omega(2^n)$
    - $\lg(n!) \ = \ \Theta(n \lg n)$

Practice Problems

- Prove or disprove the following:
  a. $4+2n+3n^2 \in O(n^2)$
  b. $2+4+6+8+...+2n+2n^2 \in O(n^2)$
  c. $2^{2n} = O(2^n)$

Will be solved in Q&A session

Thanks to contributors

Mr. Pham Van Nguyen (2022)

Dr. Thien-Binh Dang (2017 – 2022)

Prof. Hyunseung Choo (2001 – 2022)