COMP3600/6466 – Algorithms
Asymptotic Analysis
[CLRS sec. 3.1]

Hanna Kurniawati

comp_3600_6466@anu.edu.au

Australian National University | School of Computing

Topics

- What is asymptotic analysis?
- The five asymptotic notations
- Example on Insertion Sort
- Additional notes on asymptotic notations
- Using asymptotic notations to analyse algorithms

What is Asymptotic Analysis?

- In general: Asymptotic analysis means understanding the behavior of a function in the limit
- In this class: Asymptotic analysis means understanding how the running time (or memory consumption) of an algorithm increases with the size of the input *in the limit* –that is, when the input size increases without bound

Asymptotic Analysis of Algorithms

- Recall: An algorithm transforms input to output
- Intuitively, asymptotic analysis of an algorithm means finding a function that maps input size to the required computational resources (time/memory) for the algorithm to transform the input to the desired output
- In this analysis, we are interested in the trend of the growth in computational resources rather than the exact function
  - They are easier to work with & sufficient
- Therefore, the functions we are interested in this analysis are essentially bounds of the actual requirements
  - Upper bound, lower bound, upper & lower bound

Why bother?

- In general, algorithms and computer programs are **not** for a single input, but for solving problems with varying input
- Asymptotic analysis helps in:
  - Deciding which algorithms are better for solving a specific problem
  - Predicting time & memory requirements as input size grow. This could relate to the computers we need to purchase (the highest spec is not always the best, esp. when starting a business w. limited funding)

Topics

- ✓ What is asymptotic analysis?
- The five asymptotic notations
- Example on Insertion Sort
- Additional notes on asymptotic notations
- Using asymptotic notations to analyse algorithms

Five asymptotic notations

- Big-Oh (upper bound): $f(n) = O\big(g(n)\big)$
- Little-Oh (strict upper bound): $f(n) = o\big(g(n)\big)$
- Big-Omega (lower bound): $f(n) = \Omega\big(g(n)\big)$
- Little-Omega(strict lower bound): $f(n) = \omega\big(g(n)\big)$
- Theta (upper & lower bound): $f(n) = \Theta\big(g(n)\big)$

- What does the notation mean? Let's take big-oh example
  - $O\big(g(n)\big)$ is a set of functions that satisfy certain characteristics related to $g(n)$
  - $f(n) = O\big(g(n)\big)$ means $f(n) \in O\big(g(n)\big)$, and call $g(n)$ to be an asymptotic bound of $f(n)$. In the case of big-oh, asymptotic upper bound

Asymptotic notations: Big-Oh $O\big(g(n)\big)$

Def.:

$$O\big(g(n)\big) = \{f(n) \mid \exists\ positive\ constants\ c\ and\ n_0 \\ s.t.\, 0 \le f(n) \le cg(n)\ for\ \forall\ n \ge n_0\}$$

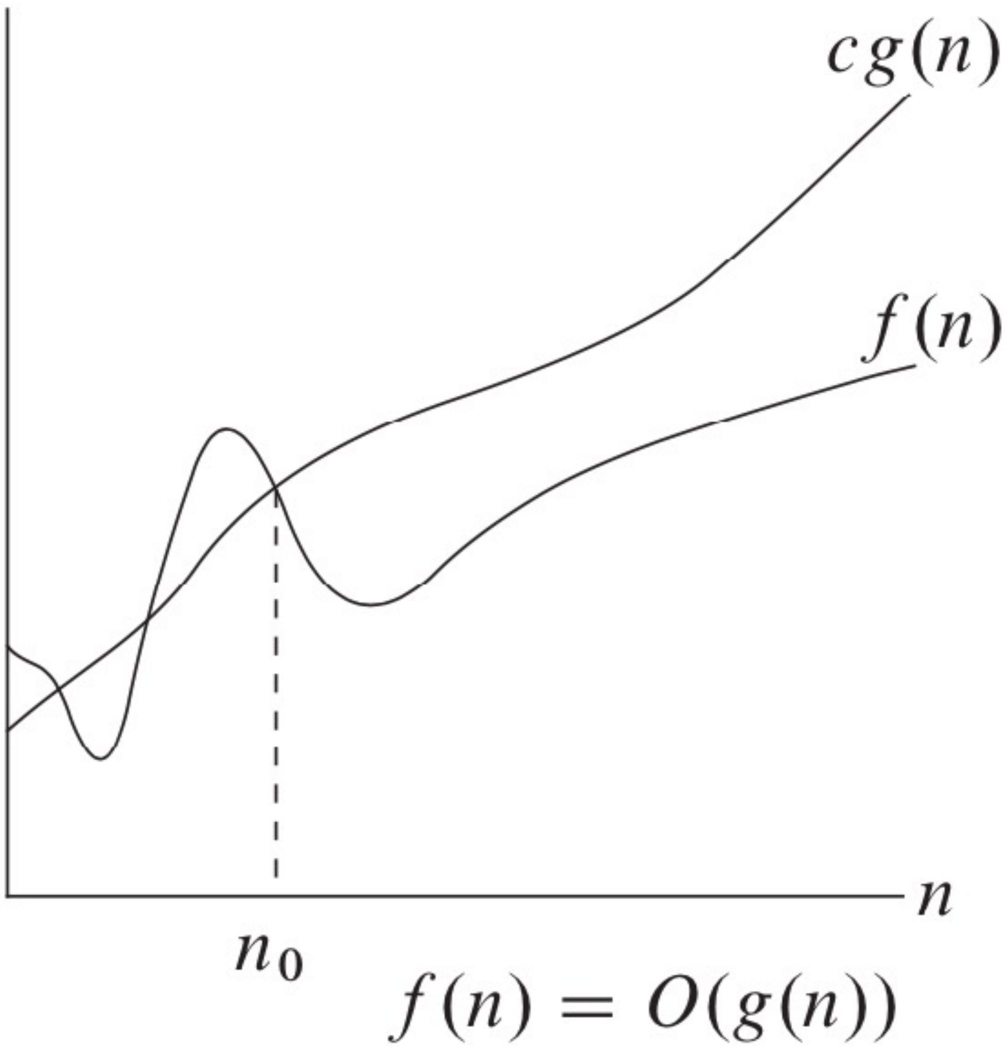
Intuitively:

$O\big(g(n)\big)$ is a set of functions that is upper bounded by a constant multiplication of $g(n)$ for "large" n –that is, whenever $n \ge n_0$.

We write $f(n) = O\big(g(n)\big)$ to mean $f(n) \in O\big(g(n)\big)$, and call $g(n)$ to be an asymptotic upper bound of $f(n)$

This bound can be tight or not tight. The function $g(n)$ is an asymptotically tight upper bound of $f(n)$ whenever for $\forall\ n \ge n_0$, $f(n)$ is equal to $g(n)$ to within a constant factor

Visually,



Taken from: [CLRS] Figure 3.1

An Example: Back to Insertion Sort

- Recall Insertion Sort and its time complexity

```
InsertionSort(A)
for j = 2 to A.length
    Key = A[j]
    i = j-1
    While i > 0 and A[i] > key
        A[i+1] = A[i]
        i = i-1
    A[i+1] = key
```

Time complexity of Insertion Sort: $\boxed{T(n) = Cn^2 + C'n - C''}$

$f(n)$

Is this information useful?

- The running-time complexity of an algorithm is at least $O(n^2)$

Membership test for Big-Oh $O\big(g(n)\big)$

- How to figure out if $f(n) \in O\big(g(n)\big)$?
  - Find the constants c and n_0
  - Use limit
    - $f(n) \in O\big(g(n)\big)$ means $f(n)$ has smaller or equal growth rate, compared to $g(n)$
    - $f(n)$ has smaller growth rate than $g(n)$ means $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
    - $f(n)$ has the same growth rate as $g(n)$ means $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c'$ for a positive constant c'
  - Examples:

    $2n \ ?\ O(n)$; $2n \ ?\ O(n^2)$; $2n^2 \ ?\ O(n)$; $2n^2 \ ?\ O(n^2)$

Just in case…

- How to compute the limit?
  - $\lim_{n \to \infty} \frac{n+1}{n^2+1} = \frac{\infty}{\infty} = undefined!!!$
  - Recall L'Hopital's rule: $\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{f'(n)}{g'(n)}$

Examples

- Is $8^n = O(2^n)$?

- Is $log(2n) = O\left(ln(n)\right)$?

Asymptotic notations: Little-Oh $o\big(g(n)\big)$

Def.:

$$o\big(g(n)\big) = \{f(n) \,|for\ any\ positive\ constant\ c > 0,$$
$$\exists\ a\ constant\ n_0 > 0\ \ s.t. 0 \le f(n) < cg(n)\ for\ \forall\ n \ge n_0\}$$

Intuitively:

$o\big(g(n)\big)$ is a set of functions that is strictly upper bounded by a constant multiplication of $g(n)$ for "large" n –that is, whenever $n \ge n_0$– in a non tight manner

We write $f(n) = o\big(g(n)\big)$ to mean $f(n) \in o\big(g(n)\big)$, and call $g(n)$ to be a non asymptotically tight upper bound of $f(n)$

Membership test for Little-Oh $o(g(n))$

- How to figure out if $f(n) \in o(g(n))$?
  - Find the constants c and n_0
  - Use limit
    - $f(n) \in o(g(n))$ means $f(n)$ has smaller growth rate compared to $g(n)$, i.e., $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
  - Example:

$$2n \;?\; o(n)\;;\; 2n \;?\; o(n^2)\;;\; 2n^2 \;?\; o(n)\;;\; 2n^2 \;?\; o(n^2)$$

For comparison

$$2n = O(n)\;;\; 2n = O(n^2)\;;\; 2n^2 \neq O(n)\;;\; 2n^2 = O(n^2)$$

Asymptotic notations: Big-Omega $\Omega\big(g(n)\big)$

Def.:

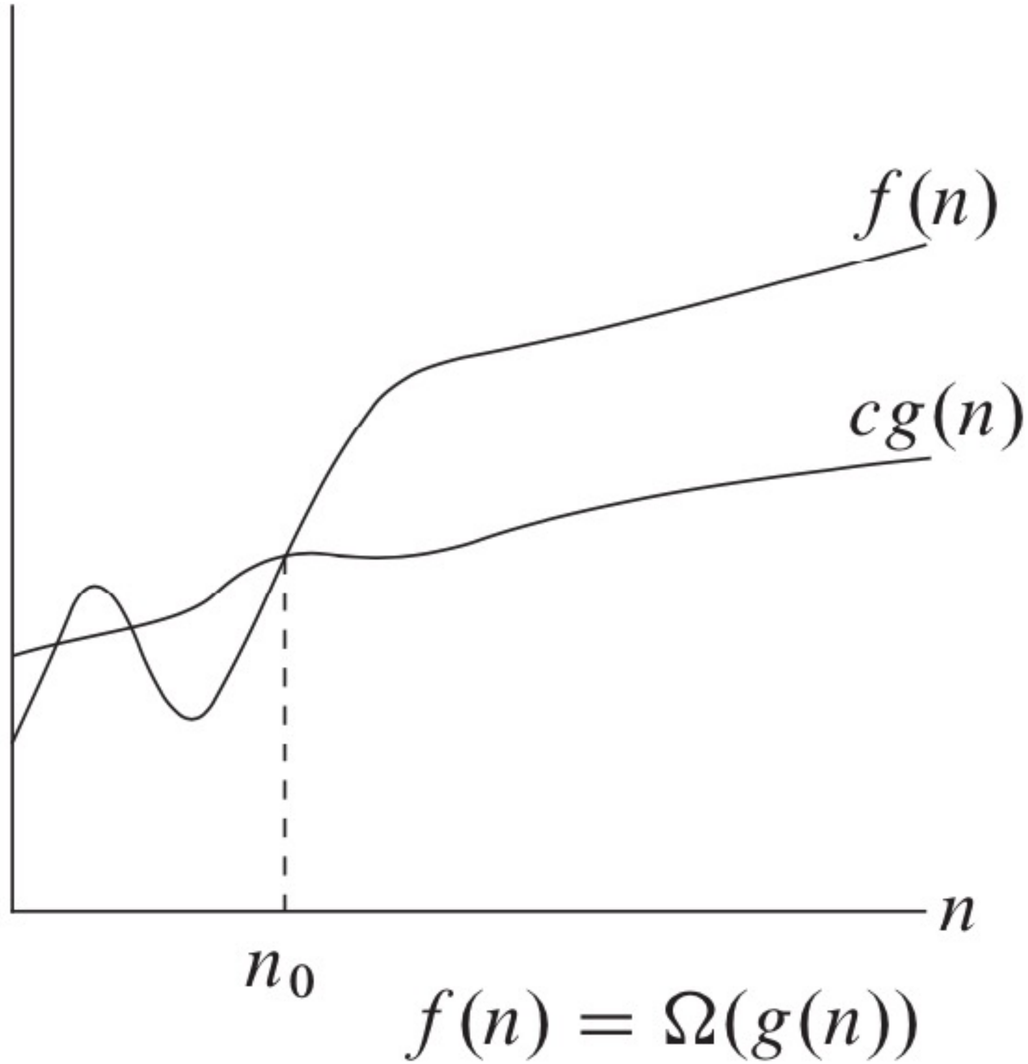
$$\Omega\big(g(n)\big) = \{f(n) \mid \exists\ positive\ constants\ c\ and\ n_0 \\ s.t.\ 0 \le cg(n) \le f(n)\ for\ \forall\ n \ge n_0\}$$

Intuitively:

$\Omega\big(g(n)\big)$ is a set of functions that is lower bounded by a constant multiplication of $g(n)$ for "large" n –that is, whenever $n \ge n_0$.

We write $f(n) = \Omega\big(g(n)\big)$ to mean $f(n) \in \Omega\big(g(n)\big)$, and call $g(n)$ to be an asymptotic lower bound of $f(n)$

Visually,



Membership test for Big-Omega $\Omega\big(g(n)\big)$

- How to figure out if $f(n) \in \Omega\big(g(n)\big)$?
  - Find the constants c and n_0
  - Use limit
    - $f(n) \in \Omega\big(g(n)\big)$ means $f(n)$ has larger or equal growth rate, compared to $g(n)$
    - $f(n)$ has larger growth rate than $g(n)$ means $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$
    - $f(n)$ has the same growth rate as $g(n)$ means $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c'$ for a positive constant c'
  - Example:

$$2n \text{ ? } \Omega(n)\,;\ 2n \text{ ? } \Omega(n^2);\ 2n^2 \text{ ? } \Omega(n)\,;\ 2n^2 \text{ ? } \Omega(n^2)$$

For comparison

$$2n = O(n)\ ;\ 2n = O(n^2)\ ;\ 2n^2 \neq O(n)\ ;\ 2n^2 = O(n^2)$$

Asymptotic notations: Little-Omega $\omega\big(g(n)\big)$

Def.:

$$\omega\big(g(n)\big) = \{f(n)\ | for\ any\ positive\ constant\ c > 0,$$
$$\exists\ a\ constant\ n_0 > 0\ s.t.\ 0 \le cg(n) < f(n)\ for\ \forall\ n \ge n_0\}$$

Intuitively:

$\omega\left(g(n)\right)$ is a set of functions that is lower bounded by a constant multiplication of $g(n)$ for large" n –that is, whenever $n \ge n_0$– in a non tight manner

We write $f(n) = \omega\left(g(n)\right)$ to mean $f(n) \in \omega\left(g(n)\right)$, and call $g(n)$ to be a non asymptotically tight lower bound of $f(n)$

Membership test for Little-Omega $\omega\big(g(n)\big)$

- How to figure out if $f(n) \in \omega\big(g(n)\big)$?
  - Find the constants c and n_0
  - Use limit
    - $f(n) \in \omega\big(g(n)\big)$ means $f(n)$ has larger growth rate compared to $g(n)$, i.e., $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$
  - Example:

$$2n \,?\, \omega(n)\ ;\ 2n \,?\, \omega(n^2)\ ;\ 2n^2 \,?\, \omega(n)\ ;\ 2n^2 \,?\, \omega(n^2)$$

For comparison

$$2n = O(n)\ ;\ 2n = O(n^2)\ ;\ 2n^2 \neq O(n)\ ;\ 2n^2 = O(n^2)$$

Asymptotic notations: Big-Theta $\Theta\big(g(n)\big)$

Def.:

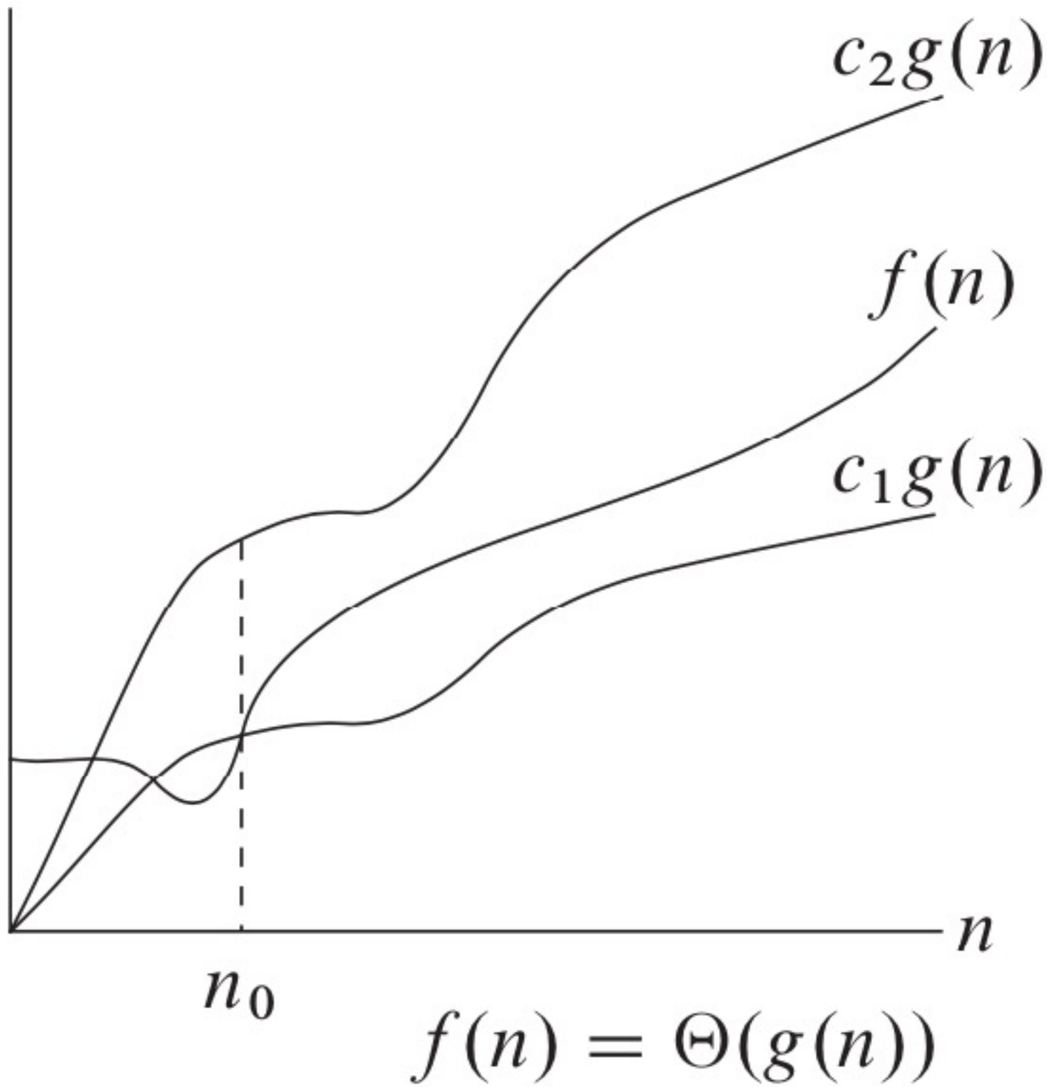
$$\Theta\big(g(n)\big) = \{f(n) \mid \exists\ positive\ constants\ c_1, c_2, and\ n_0$$
$$s.t.\ 0 \le c_1 g(n) \le f(n) \le c_2 g(n)\ for\ \forall\ n \ge n_0\}$$

Intuitively:

$\Theta\big(g(n)\big)$ is a set of functions that is asymptotically upper and lower bounded by constants multiplication of $g(n)$ for "large" n –that is, whenever $n \ge n_0$.

We write $f(n) = \Theta\big(g(n)\big)$ to mean $f(n) \in \Theta\big(g(n)\big)$, and call $g(n)$ to be an asymptotically tight bound of $f(n)$

Visually,



Membership test for Big-Theta $\Theta(g(n))$

- How to figure out if $f(n) \in \Theta(g(n))$?
  - Find the constants c_1, c_2, and n_0
  - Use limit
    - $f(n) \in \Theta(g(n))$ means $f(n)$ has the same growth rate as $g(n)$, i.e., $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c'$ for a positive c'
  - Example:

    $2n \ ?\ \Theta(n)$; $2n \ ?\ \Theta(n^2)$; $2n^2 \ ?\ \Theta(n)$; $2n^2 \ ?\ \Theta(n^2)$

  For comparison

    $2n = O(n)$; $2n = O(n^2)$; $2n^2 \neq O(n)$; $2n^2 = O(n^2)$

Asymptotic notations: Little-Theta?

- Actually, little-theta does not exist. And the previous notation is usually called “Theta” rather than “Big-Theta”
  - Why?

Summary of Asymptotic notations

Asymptotic bounds of $\boldsymbol{f(n)}$	Upper bound	Lower bound
May be tight or not / inclusive	$O\big(g(n)\big)$	$\Omega\big(g(n)\big)$
Non-tight / strict	$o\big(g(n)\big)$	$\omega\big(g(n)\big)$
Tight	$\Theta\big(g(n)\big)$	

- These notations are essentially sets of functions
- Membership test:
  - Finding the appropriate constants
  - Limit definition

Summary of Membership Tests for Asymptotic notations

Asymptotic bounds of $f(n)$ & membership test based on limit	Upper bound	Lower bound
May be tight or not / inclusive	$O(g(n))$ $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$ or c'	$\Omega(g(n))$ $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$ or c'
Non-tight / strict	$o(g(n))$ $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$	$\omega(g(n))$ $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$
Tight	$\Theta(g(n))$ $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c'$	

c' : Positive constant

Topics

- ✓ What is asymptotic analysis?
- ✓ The five asymptotic notations
- Example on Insertion Sort
- Additional notes on asymptotic notations
- Using asymptotic notations to analyse algorithms

Asymptotic Analysis of Algorithms: Back to Insertion Sort Example

- Recall Insertion Sort and its time complexity

```
InsertionSort(A)
1.  for j = 2 to A.length
2.      Key = A[j]
3.      i = j-1
4.      While i > 0 and A[i] > key
5.          A[i+1] = A[i]
6.          i = i-1
7.      A[i+1] = key
```

Time complexity of Insertion Sort: $T(n) = Cn^2 + C'n - C''$

Tight asymptotic bound of the running time of Insertion Sort: ?

Topics

✓What is asymptotic analysis?

✓The five asymptotic notations

✓Example on Insertion Sort

To be continued next Monday…

- Additional notes on asymptotic notations
- Using asymptotic notations to analyse algorithms