3.1 Asymptotic notation

The notations used to describe the asymptotic running time of an algorithm are defined in terms of functions whose domain is the set of natural numbers.

- O-notation.

  For a given function $g(n) \geq 0$, denote by $O(g(n))$ the following set of functions.

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

  $f(n) = O(g(n))$ means function $g(n)$ is an asymptotically upper bound for $f(n)$.

  E.g., Given $f(n) = 3.5n - 78$ and $g(n) = (1/2)n^2$, then, $f(n) = O(g(n))$.
  Given $f(n) = n^3$ and $g(n) = 2n^2$, then, $f(n) \neq O(g(n))$.

  Exercise: Given $f(n) = n^{3/2}$ and $g(n) = 2n^2$, then, $f(n) = O(g(n))$.

3.1 Asymptotic notation (cont.)

- Ω-notation.

  For a given function $g(n)$, denote by $\Omega(g(n))$ the following set of functions.

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

  $f(n) = \Omega(g(n))$ means $g(n)$ is an asymptotically lower bound of $f(n)$.

  E.g., Given $f(n) = 3.5n^2 + 5n - 3$ and $g(n) = n$, then, $f(n) = \Omega(g(n))$.
  Given $f(n) = n \log n + 10n$ and $g(n) = \log n$, then, $f(n) = \Omega(g(n))$.

  Exercise: Given $f(n) = n^2$ and $g(n) = 4n^3$, whether $f(n) = \Omega(g(n))$?

3.1 Asymptotic notation (cont.)

- Θ-notation.

  For a given $g(n)$, denote by $\Theta(g(n))$ the following set of functions.

  $\Theta(g(n)) = \{ f(n)$: there **exist** positive constants c_1, c_2 and n_0 such that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for **all** $n \geq n_0\}$.

  $f(n) = \Theta(g(n))$ means function $f(n)$ is equal to $g(n)$ to within a constant factor, and $g(n)$ is an asymptotically tight bound for $f(n)$.

  E.g., Given $f(n) = 1/2n^2 - 3n + 7$ and $g(n) = n^2$, then $f(n) = \Theta(g(n)) = \Theta(n^2)$.

  Exercise: Given $f(n) = \sum_{i=1}^{n} i$ and $g(n) = n^2$, then $f(n) = \Theta(g(n)) = \Theta(n^2)$.

3.1 Asymptotic notation (cont.)

- o-notation.

  o-notation is used to denote an upper bound that is not **asymptotically tight**.

  For a given function $g(n)$, denote by $o(g(n))$ the following set of functions.

  $o(g(n)) = \{ f(n)$: for **any** positive constant c, there **exists** a constant n_0 such that $0 \leq f(n) < cg(n)$ for **all** $n \geq n_0\}$.

  If $g(n) > 0$ for large enough n, this is equivalent to

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

  E.g., $2n = o(n^2)$ but $2n^2 \neq o(n^2)$.

  Exercise: $\frac{3}{5}\sqrt{n} = o(n)$?.

3.1 Asymptotic notation (cont.)

- ω-notation.

  ω-notation is used to denote a lower bound that is not asymptotically tight. For a given function $g(n)$, denote by $\omega(g(n))$ the following set of functions.

  $\omega(g(n)) = \{ f(n)$: for **any** positive constant c there **exists** a constant n_0 such that $0 \leq cg(n) < f(n)$ for **all** $n \geq n_0\}$.

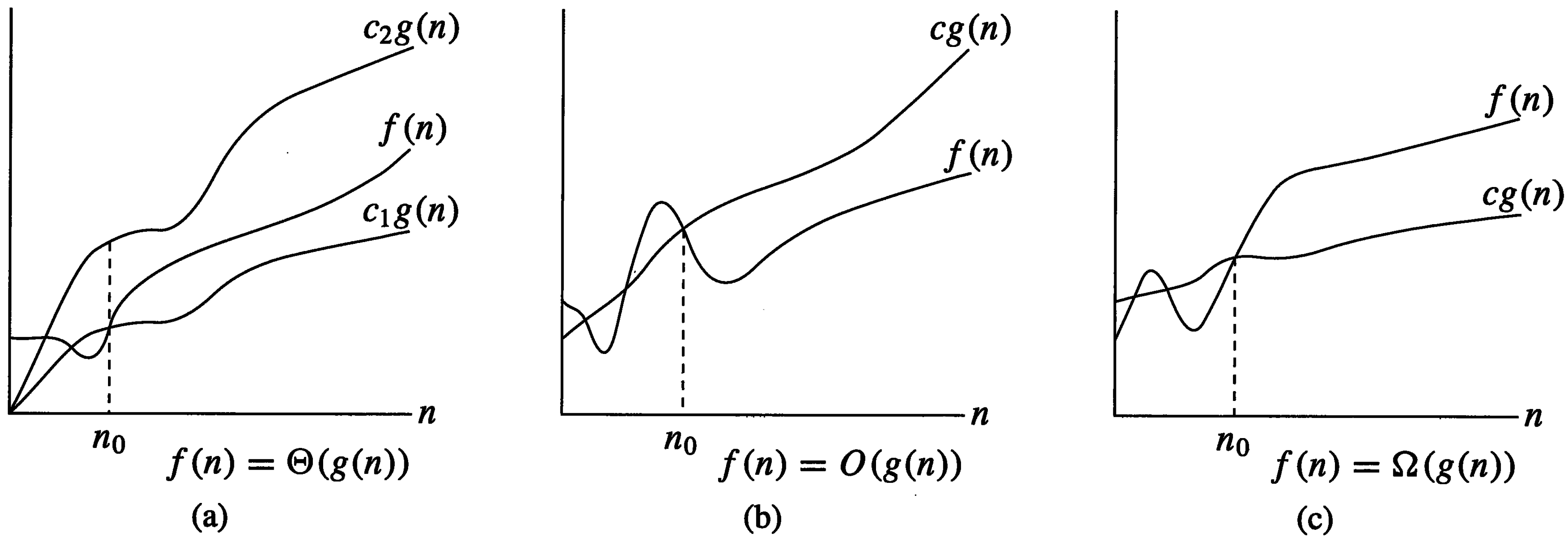
  If $g(n) > 0$ for large enough n, this is equivalent to

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

  E.g., $n^2/2 = \omega(n)$ but $n^2/2 \neq \omega(n^2)$.

  Exercise: $n^3 = \omega(\sqrt{n})$?

3.1 Asymptotic notation (cont.)



Cormen, p43

Why do we need n_0?

Consider the definition of $f(n) = O(g(n))$.

The purpose of the $n \geq n_0$ condition is to avoid inconvenient behaviour for tiny n.

One example is when $f(n)$ is negative for tiny n.

The most important inconvenient behaviour in practice is that $g(n)$ might be 0.

Example: $1 + 2\ln n = O(\ln n)$

This statement is true, but there is no constant c such that $0 \leq 1 + 2\ln n \leq c\ln n$ when $n = 1$. This is not very interesting, so we use $n_0 = 2$ to cut off the case $n = 1$:

$$0 \leq 1 + 2\ln n \leq 4\ln n \quad \text{for} \quad n \geq 2.$$

Exercise: If $f(n) = O(g(n))$ and $f(n), g(n) > 0$ for $n \geq 1$, then there is a constant c such that $0 \leq f(n) \leq cg(n)$ for $n \geq 1$.

The strange behaviour of equations with $O()$

Usually $x = y$ implies $y = x$. However, $f(n) = O(g(n))$ does not imply $O(g(n)) = f(n)$.

Similarly $O(n) = O(n^2)$ but $O(n^2) \neq O(n)$.

The explanation is that "=" has a different meaning when $O()$ appears, and similarly for $\Theta()$ and $\Omega()$. Remember that $O(g(n))$ represents a **set** of functions.

The meaning of "=" is:
Any function that belongs to the left side also belongs to the right side.

So the meaning is very similar to "subset".

Consider: $n + n^2 O(\ln n) = O(n^2 \ln n)$.