Asymptotic Notation

- What is Asymptotic?

Asymptotic Notation

- What is Asymptotic?
  - If something is asymptotic, it means that its nature, is of an asymptote.

Asymptotic Notation

- What is Asymptotic?
  - If something is asymptotic, it means that its nature, is of an asymptote.
- What is an Asymptote?

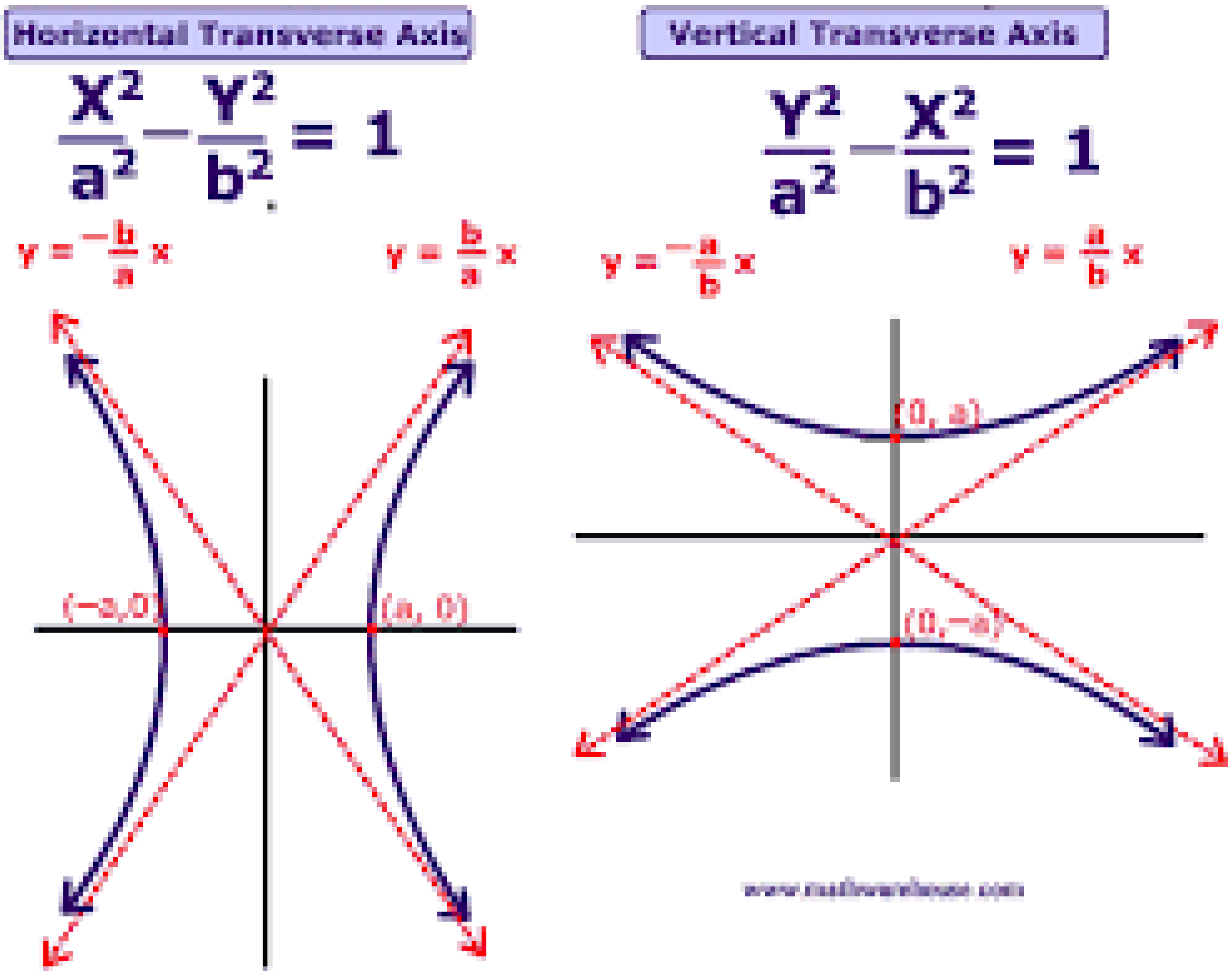
Asymptotic Notation

- What is Asymptotic?
  - If something is asymptotic, it means that its nature, is of an asymptote.

- What is an Asymptote?
  - A straight line that is the limiting value of a curve; can be considered as tangent at infinity.

Asymptotic Notation

➢



Asymptotic Notation

- What is Asymptotic?
  - If something is asymptotic, it means that its nature, is of an asymptote.
- What is an Asymptote?
  - A straight line that is the limiting value of a curve; can be considered as tangent at infinity.
- What is asymptotic efficiency (analysis) of an algorithm?

Asymptotic Notation

- What is Asymptotic?
  - If something is asymptotic, it means that its nature, is of an asymptote.
- What is an Asymptote?
  - A straight line that is the limiting value of a curve; can be considered as tangent at infinity.
- What is asymptotic efficiency (analysis) of an algorithm?
  - In asymptotic efficiency of an algorithm, the concern is how the running time of an algorithm increases with the size of an input in the limit, as the size of the input increases without bound.

Asymptotic Notation

- What is Asymptotic?
  - If something is asymptotic, it means that its nature, is of an asymptote.

- What is an Asymptote?
  - A straight line that is the limiting value of a curve; can be considered as tangent at infinity.

- What is asymptotic efficiency (analysis) of an algorithm?
  - In asymptotic efficiency of an algorithm, the concern is how the running time of an algorithm increases with the size of an input in the limit, as the size of the input increases without bound.

- Usually, an algorithm that is asymptotically more efficient will be the best choice for all but very small inputs.

Asymptotic Notation

- What is Asymptotic?
  - If something is asymptotic, it means that its nature, is of an asymptote.

- What is an Asymptote?
  - A straight line that is the limiting value of a curve; can be considered as tangent at infinity.

- What is asymptotic efficiency (analysis) of an algorithm?
  - In asymptotic efficiency of an algorithm, the concern is how the running time of an algorithm increases with the size of an input in the limit, as the size of the input increases without bound.

- Usually, an algorithm that is asymptotically more efficient will be the best choice for all but very small inputs.

- We will use asymptotic notation primarily to describe the running times of algorithms.

Asymptotic Notation

- Which are different asymptotic notations?
  - Θ**-notation**
  - O**-notation**
  - Ω**-notation**
  - o**-notation**
  - ω**-notation**

Asymptotic Notation

- The notations we use to describe the asymptotic running time of an algorithm are **defined in terms of functions whose domains are the set of natural numbers N = {0, 1, 2, …}.**

Asymptotic Notation

- **Θ-notation**

For a given function $g(n)$, we denote by $\Theta(g(n))$ the *set of functions*

$$\Theta(g(n)) = \{ f(n) : \text{ there exist positive constants } c_1, c_2, \text{ and } n_0 \text{ such that } 0 \le c_1 g(n) \le f(n) \le c_2 g(n) \text{ for all } n \ge n_0 \} .$$[1]

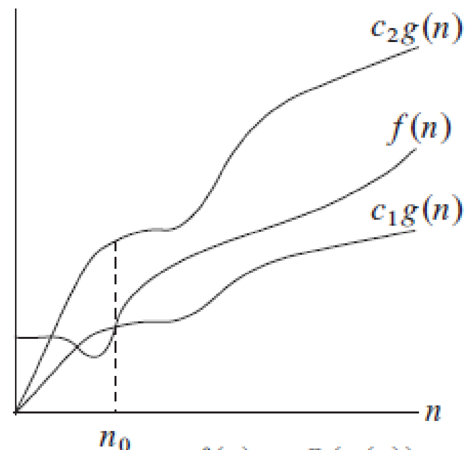


Figure 3.1 Graphic examples of the Θ, O, and Ω notations.

In each part, the value of n_0 shown is the minimum possible value; any greater value would also work. **(a)** Θ-notation bounds a function to within constant factors. We write $f(n) = \Theta(g(n))$ if there exist positive constants n_0, c_1, and c_2 such that at and to the right of n_0, the value of $f(n)$ always lies between $c_1 g(n)$ and $c_2 g(n)$ inclusive.

Asymptotic Notation

- **Θ-notation**
  - A function $f(n)$ belongs to the set $\Theta(g(n))$ if there exist positive constants c_1 and c_2 such that it can be "sandwiched" between $c_1 g(n)$ and $c_2 g(n)$, for sufficiently large n.

Asymptotic Notation

- **Θ-notation**
  - A function $f(n)$ belongs to the set $\Theta(g(n))$ if there exist positive constants c_1 and c_2 such that it can be "sandwiched" between $c_1 g(n)$ and $c_2 g(n)$, for sufficiently large n.
  - Because $\Theta(g(n))$ is a set, we could write "$f(n) \in \Theta(g(n))$" to indicate that $f(n)$ is a member of $\Theta(g(n))$.

Asymptotic Notation

- **Θ-notation**
  - A function $f(n)$ belongs to the set $\Theta(g(n))$ if there exist positive constants c_1 and c_2 such that it can be "sandwiched" between $c_1 g(n)$ and $c_2 g(n)$, for sufficiently large n.
  - Because $\Theta(g(n))$ is a set, we could write "$f(n) \in \Theta(g(n))$" to indicate that $f(n)$ is a member of $\Theta(g(n))$.
  - Instead, we will usually write "$f(n) = \Theta(g(n))$" to express the same notion.

Asymptotic Notation

- **Θ-notation**
  - A function $f(n)$ belongs to the set $\Theta(g(n))$ if there exist positive constants c_1 and c_2 such that it can be "sandwiched" between $c_1 g(n)$ and $c_2 g(n)$, for sufficiently large n.
  - Because $\Theta(g(n))$ is a set, we could write "$f(n) \in \Theta(g(n))$" to indicate that $f(n)$ is a member of $\Theta(g(n))$.
  - Instead, we will usually write "$f(n) = \Theta(g(n))$" to express the same notion.
  - You might be confused because we abuse equality in this way, but we shall see later in this section that doing so has its advantages.

Asymptotic Notation

- **Θ-notation**
  - Figure 3.1(a) gives an intuitive picture of functions f(n) and g(n), where $f(n) = \theta(g(n))$.

Asymptotic Notation

- **Θ-notation**
  - Figure 3.1(a) gives an intuitive picture of functions f(n) and g(n), where $f(n) = \theta(g(n))$.
  - For all values of n at and to the right of n_0, the value of f(n) lies at or above $c_1g(n)$ and at or below $c_2g(n)$. In other words, for all $n \geq n_0$, the function f(n) is equal to g(n) to within a constant factor. We say that g(n) is an asymptotically tight bound for f(n).

Asymptotic Notation

- **Θ-notation**

In Chapter 2, we introduced an informal notion of Θ-notation that amounted to throwing away lower-order terms and ignoring the leading coefficient of the highest-order term. Let us briefly justify this intuition by using the formal definition to show that $\frac{1}{2}n^2 - 3n = \Theta(n^2)$. To do so, we must determine positive constants c_1, c_2, and n_0 such that

$$c_1 n^2 \le \frac{1}{2}n^2 - 3n \le c_2 n^2$$

for all $n \ge n_0$. Dividing by n^2 yields

$$c_1 \le \frac{1}{2} - \frac{3}{n} \le c_2 \,.$$

n	f(n)		n	f(n)
1	-5/2		6	0
2	-1		7	1/14
3	-1/2		8	1/8
4	-1/4		9	1/6
5	-1/10		10	1/5

We can make the right-hand inequality hold for any value of $n \ge 1$ by choosing any constant $c_2 \ge 1/2$. Likewise, we can make the left-hand inequality hold for any value of $n \ge 7$ by choosing any constant $c_1 \le 1/14$. Thus, by choosing $c_1 = 1/14$, $c_2 = 1/2$, and $n_0 = 7$, we can verify that $\frac{1}{2}n^2 - 3n = \Theta(n^2)$. Certainly, other choices for the constants exist, but the important thing is that *some* choice exists. Note that these constants depend on the function $\frac{1}{2}n^2 - 3n$; a different function belonging to $\Theta(n^2)$ would usually require different constants.

Asymptotic Notation

- **Θ-notation**

We can also use the formal definition to verify that $6n^3 \neq \Theta(n^2)$. Suppose for the purpose of contradiction that c_2 and n_0 exist such that $6n^3 \leq c_2 n^2$ for all $n \geq n_0$. But then dividing by n^2 yields $n \leq c_2/6$, which cannot possibly hold for arbitrarily large n, since c_2 is constant.

Asymptotic Notation

- **Θ-notation**

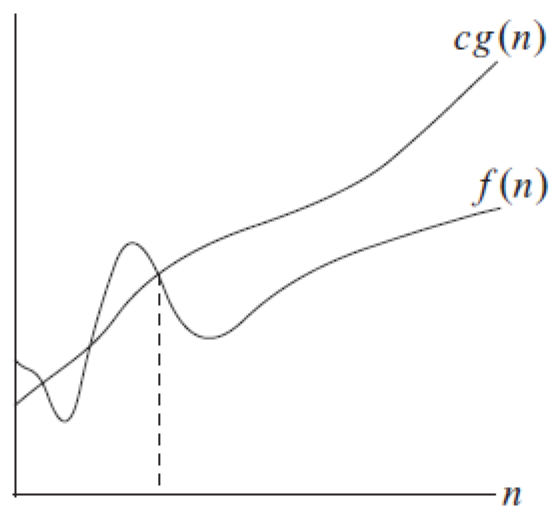
Since any constant is a degree-0 polynomial, we can express any constant function as $\Theta(n^0)$, or $\Theta(1)$. This latter notation is a minor abuse, however, because the expression does not indicate what variable is tending to infinity.[2] We shall often use the notation $\Theta(1)$ to mean either a constant or a constant function with respect to some variable.

Asymptotic Notation

- O**-notation**

The Θ-notation asymptotically bounds a function from above and below. When we have only an ***asymptotic upper bound***, we use O-notation. For a given function $g(n)$, we denote by $O(g(n))$ (pronounced "big-oh of g of n" or sometimes just "oh of g of n") the set of functions

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



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

Asymptotic Notation

- **O-notation**
  - f(n) = $O(g(n))$ means that a function f(n) belongs to the set $O(g(n))$.

Asymptotic Notation

- ***O*-notation**
  - f(n) = O(g(n)) means that a function f(n) belongs to the set O(g(n)).
  - $f(n) = \theta\big(g(n)\big)$ implies f(n) = O(g(n)), that is $\theta\big(g(n)\big) \subseteq O\big(g(n)\big)$

Asymptotic Notation

- ***O*-notation**
  - f(n) = O(g(n)) means that a function f(n) belongs to the set O(g(n)).
  - $f(n) = \theta\big(g(n)\big)$ implies f(n) = O(g(n)), that is $\theta\big(g(n)\big) \subseteq O\big(g(n)\big)$
  - So, any quadratic function $an^2 + bn + c$, where a > 0 is not only in $\theta(n^2)$ but also in $O(n^2)$.

Asymptotic Notation

- **O-notation**
  - f(n) = O(g(n)) means that a function f(n) belongs to the set O(g(n)).
  - $f(n) = \theta(g(n))$ implies f(n) = O(g(n)), that is $\theta(g(n)) \subseteq O(g(n))$
  - So, any quadratic function $an^2 + bn + c$, where a > 0 is not only in $\theta(n^2)$ but also in $O(n^2)$.
  - It may be surprising that when a > 0, any linear function an + b is in $O(n^2)$. This can be verified by taking c = a + |b| and n_0 = max (1, -b/a).

Asymptotic Notation

- ***O*-notation**
  - f(n) = O(g(n)) means that a function f(n) belongs to the set O(g(n)).
  - $f(n) = \theta(g(n))$ implies f(n) = O(g(n)), that is $\theta(g(n)) \subseteq O(g(n))$
  - So, any quadratic function $an^2 + bn + c$, where $a > 0$ is not only in $\theta(n^2)$ but also in $O(n^2)$.
  - It may be surprising that when $a > 0$, any linear function $an + b$ is in $O(n^2)$. This can be verified by taking $c = a + |b|$ and $n_0 = \max(1, -b/a)$.
  - When we write f(n) = O(g(n)), we are merely claiming that some constant multiple of g(n) is an asymptotic upper bound on f(n), with no claim of how tight an upper bound it is.

Asymptotic Notation

- ***O*-notation**
  - Using O-notation, we can often describe the running time of an algorithm merely by inspecting the algorithm's overall structure.

Asymptotic Notation

- ***O*-notation**
  - Using O-notation, we can often describe the running time of an algorithm merely by inspecting the algorithm's overall structure.
  - For example, the doubly nested loop structure of the insertion sort algorithm from Chapter 2 immediately yields an $O(n^2)$ upper bound on the worst-case running time: the cost of each iteration of the inner loop is bounded from above by $O(1)$ (constant), the indices i and j are both at most n, and the inner loop is executed at most once for each of the n^2 pairs of values for i and j .

Asymptotic Notation

- ***O*-notation**

Since O-notation describes an upper bound, when we use it to bound the worst-case running time of an algorithm, we have a bound on the running time of the algorithm on every input—the blanket statement we discussed earlier.

Asymptotic Notation

- ***O*-notation**

Since O-notation describes an upper bound, when we use it to bound the worst-case running time of an algorithm, we have a bound on the running time of the algorithm on every input—the blanket statement we discussed earlier.

Thus, the $O(n^2)$ bound on worst-case running time of insertion sort also applies to its running time on every input.

Asymptotic Notation

- ***O*-notation**

Since O-notation describes an upper bound, when we use it to bound the worst-case running time of an algorithm, we have a bound on the running time of the algorithm on every input—the blanket statement we discussed earlier.

Thus, the $O(n^2)$ bound on worst-case running time of insertion sort also applies to its running time on every input.

The $\Theta(n^2)$ bound on the worst-case running time of insertion sort, however, does not imply a $\Theta(n^2)$ bound on the running time of insertion sort on *every* input. For example, we saw in Chapter 2 that when the input is already sorted, insertion sort runs in $\Theta(n)$ time.

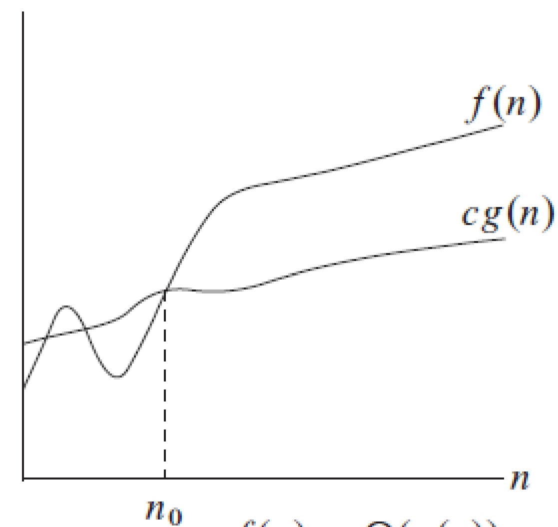
Asymptotic Notation

- **Ω-notation**

Just as O-notation provides an asymptotic *upper* bound on a function, Ω-notation provides an ***asymptotic lower bound***. For a given function $g(n)$, we denote by $\Omega(g(n))$ (pronounced "big-omega of g of n" or sometimes just "omega of g of n") the set of functions

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

Figure 3.1(c) shows the intuition behind Ω-notation. For all values n at or to the right of n_0, the value of $f(n)$ is on or above $cg(n)$.



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

Asymptotic Notations

- ***Theorem 3.1***

  For any two functions $f(n)$ and $g(n)$, we have $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$. ■

Asymptotic Notations

- ***Theorem 3.1***
  For any two functions $f(n)$ and $g(n)$, we have $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$. ■

- When best case running time of some algorithm is $\Omega(g(n))$, it means that running time of that algorithm for every input is also $\Omega(g(n))$

Asymptotic Notations

- Asymptotic notation in equations and inequalities

We have already seen how asymptotic notation can be used within mathematical formulas. For example, in introducing O-notation, we wrote "$n = O(n^2)$." We might also write $2n^2 + 3n + 1 = 2n^2 + \Theta(n)$. How do we interpret such formulas?

Asymptotic Notations

- Asymptotic notation in equations and inequalities

We have already seen how asymptotic notation can be used within mathematical formulas. For example, in introducing O-notation, we wrote "$n = O(n^2)$." We might also write $2n^2 + 3n + 1 = 2n^2 + \Theta(n)$. How do we interpret such formulas?

When the asymptotic notation stands alone (that is, not within a larger formula) on the right-hand side of an equation (or inequality), as in $n = O(n^2)$, we have already defined the equal sign to mean set membership: $n \in O(n^2)$. In general, however, when asymptotic notation appears in a formula, we interpret it as standing for some anonymous function that we do not care to name. For example, the formula $2n^2 + 3n + 1 = 2n^2 + \Theta(n)$ means that $2n^2 + 3n + 1 = 2n^2 + f(n)$, where $f(n)$ is some function in the set $\Theta(n)$. In this case, we let $f(n) = 3n + 1$, which indeed is in $\Theta(n)$.

Asymptotic Notations

- ***o*-notation**

The asymptotic upper bound provided by O-notation may or may not be asymptotically tight. The bound $2n^2 = O(n^2)$ is asymptotically tight, but the bound $2n = O(n^2)$ is not. We use o-notation to denote an upper bound that is not asymptotically tight. We formally define $o(g(n))$ ("little-oh of g of n") as the set

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

For example, $2n = o(n^2)$, but $2n^2 \neq o(n^2)$.

Asymptotic Notations

- **ω-notation**

By analogy, ω-notation is to Ω-notation as o-notation is to O-notation. We use ω-notation to denote a lower bound that is not asymptotically tight. One way to define it is by

$f(n) \in \omega(g(n))$ if and only if $g(n) \in o(f(n))$.

Formally, however, we define $\omega(g(n))$ ("little-omega of g of n") as the set

$$\omega(g(n)) = \{ f(n) : \text{for any positive constant } c > 0, \text{ there exists a constant } n_0 > 0 \text{ such that } 0 \le cg(n) < f(n) \text{ for all } n \ge n_0 \} .$$

Asymptotic Notations

- **ω-notation**

By analogy, ω-notation is to Ω-notation as o-notation is to O-notation. We use ω-notation to denote a lower bound that is not asymptotically tight. One way to define it is by

$$f(n) \in \omega(g(n)) \text{ if and only if } g(n) \in o(f(n)) .$$

Formally, however, we define $\omega(g(n))$ ("little-omega of g of n") as the set

$$\omega(g(n)) = \{ f(n) : \text{for any positive constant } c > 0 \text{, there exists a constant } n_0 > 0 \text{ such that } 0 \le cg(n) < f(n) \text{ for all } n \ge n_0 \} .$$

For example, $n^2/2 = \omega(n)$, but $n^2/2 \ne \omega(n^2)$. The relation $f(n) = \omega(g(n))$ implies that

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

if the limit exists. That is, $f(n)$ becomes arbitrarily large relative to $g(n)$ as n approaches infinity.