Standard Notations and Common Functions

Asymptotic Notation

Asymptotic efficiency of algorithm deals with how the running time increases with the size of the input. Usually, an algorithm that is asymptotically more efficient will be the best choice for all but very small inputs. Additionally, it is also important to be clear what type of running time that we are discussing - worst case, expected, best case all come with different notations.

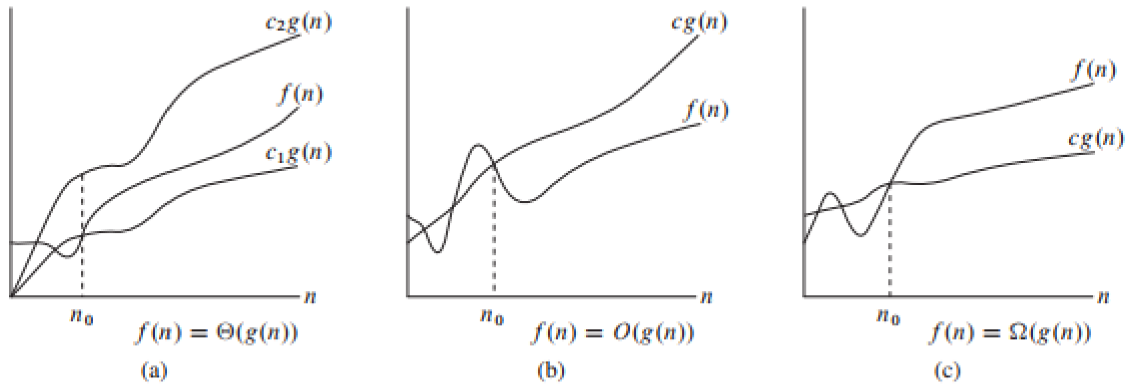

Figure 3.1 Graphic examples of the Θ, O, and Ω notations. In each part, the value of n_0 shown is the minimum possible value; any greater value would also work. **(a)** Θ-notation bounds a function to within constant factors. We write $f(n) = \Theta(g(n))$ if there exist positive constants n_0, c_1, 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. **(b)** O-notation gives an upper bound for a function to within a constant factor. We write $f(n) = O(g(n))$ if there are positive constants n_0 and c such that at and to the right of n_0, the value of $f(n)$ always lies on or below $cg(n)$. **(c)** Ω-notation gives a lower bound for a function to within a constant factor. We write $f(n) = \Omega(g(n))$ if there are positive constants n_0 and c such that at and to the right of n_0, the value of $f(n)$ always lies on or above $cg(n)$.

$$
\begin{aligned}
&\Theta(g(n)) = \{f(n) : \exists c_1 > 0, c_2 > 0, n_0 \in \mathbb{N}, \forall n \geq n_0, 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)\} \\
&O(g(n)) = \{f(n) : \exists c > 0, n_0 \in \mathbb{N}, \forall n \geq n_0, 0 \leq f(n) \leq cg(n)\} \\
&\Omega(g(n)) = \{f(n) : \exists c > 0, n_0 \in \mathbb{N}, \forall n \geq n_0, 0 \leq c_2 g(n) \leq f(n)\}
\end{aligned}
$$

Since the asymptotic notation is a set, the correct way to write running time should be

$$T(n) \in O(g(n))$$

but in general, this is usually written as $T(n) = O(g(n))$ to mean the same thing. Based on the definition of asymptotic notation, $T(n)$ is asymptotically non negative -i.e. when n reaches a certain level, $T(n)$ will become non negative. This often has a correct physical interpretation when it comes to running time of an algorithm. When $T(n) = \Theta(g(n))$, we say that $g(n)$ is an *asymptotic tight bound* for $T(n)$. When $T(n) = O(g(n))$, we say that $g(n)$ is an *asymptotic upper bound* for $T(n)$. When $T(n) = \Omega(g(n))$, we say that $g(n)$ is an *asymptotic lower bound* for $T(n)$. We usually use O to denote the worst-case running time, Ω to denote the best-case running time, and Θ to denote the average case running time of an algorithm.

Note that the bound specified by O may or may not be a tight upper-bound. For instance, $T(n) = cn$ has worst-case of $O(n)$, but it is also true to say that $T(n) = O(n^2)$. Sometimes, people use O to mean tight asymptotic bounds (find the tightest that you can prove). Based on the definition of asymptotic notations, we can drop lower order terms simply by finding a combination of n_0 and c such that the statement holds. We will show an example of this in a later exercise (excercise 3.1.2). It is also sufficient to show (see exercise 3.1.1) that the running time of an algorithm is $\Theta(g(n))$ if its worst case running time is $O(g(n))$ and its best case running time is $\Omega(g(n))$.

As mentioned previously, the use of O and Ω implies a bound that may or may not be tight. To denote a bound that is loose, we use o and ω, both of which are less commonly seen:

$$o(g(n)) = \{f(n) : \forall c > 0, \exists n_0 > 0 : \forall n \geq n_0, 0 \leq f(n) < cg(n)\}$$
$$\omega(g(n)) = \{f(n) : \forall c > 0, \exists n_0 > 0 : \forall n \geq n_0, 0 \leq cg(n) < f(n)\}$$

You make see the following definitions that are equivalent:

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

The main difference between O and o is that O only requires the statement to be true for **some** constant c, while o requires the statement to be true for all constant c, hence if $f(n) = o(g(n))$, then $f(n) = O(g(n))$, but the reverse may not be true. Intuitively, the previous statement means that if g is a loose upper bound for f, then g is an upper bound for f. You will see in later exercises that many relational properties of the real numbers also hold for asymptotic relations. We will prove them as we move along.

Prove the following claims:

- 3.1.1/ $f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) \text{ and } f(n) = \Omega(g(n))$
  Use definition
- 3.1.2/ Prove that if $T(n) = an^2 + bn + c$, $T(n) = \Theta(n^2)$ for $a, b, c \in \mathbb{R}$.
- 3.1.3/ Use the following definitions of limits to prove the limit definition of o and ω:
  - We say $\lim_{n\to\infty} f(n) = c$ if for all $\epsilon > 0$, there exists $n_0 \in \mathbb{N}$ such that for all $n > n_0$, $|f(n) - c| < \epsilon$.
  - We say $\lim_{n\to\infty} f(n) = \infty$ if for all $\epsilon > 0$, there exists $n_0 \in \mathbb{N}$ such that for all $n > n_0$, $f(n) > \epsilon$.
    Use definition again, quite straight forward.
- 3.1.4/Prove the Transitivity, Reflexivity, Symmetry, Transpose Symmetry properties on page 51,52 of the textbook.

Do exercises 3.1.1 - 3.1.7 in the textbook. (I have the solution but haven't written them down here).

Increasing and Decreasing Functions

Let $f : D \to \mathbb{R}$ be a mapping from a domain $D \subset \mathbb{R}$ to the real line,
f **monotonically increasing** if for

$$x, y \in D, x \leq y \to f(x) \leq f(y)$$

monotonically decreasing if

$$x \leq y \to f(x) \geq f(y)$$

Examples:

- $f(x) = x$ is monotonically increasing, since
  $x \leq y \to f(x) = x \leq f(y) = y$
- $f(x) = 1/x$ is monotonically decreasing, since
  $x \leq y \to 1/x \geq 1/y \to f(x) \geq f(y)$

Floors and Ceilings:

For $x \in \mathbb{R}$:

$$\lfloor x \rfloor \triangleq \max\{n \in \mathbb{Z} : n \leq x\} \triangleq \inf_{n \in \mathbb{Z}} x$$

$$\lceil x \rceil \triangleq \min\{n \in \mathbb{Z} : n \geq x\} \triangleq \sup_{n \in \mathbb{Z}} x$$

Some useful inequalities:

- 3.2.1/ $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$

- 3.2.2/ $\lfloor x/2 \rfloor + \lceil x/2 \rceil = x$ for $x \in \mathbb{Z}$
  Proof:

- When x is divisible by 2 - i.e. $x = 2k$, $\lfloor x/2 \rfloor = \lceil x/2 \rceil = k$. Hence:
  $\lfloor x/2 \rfloor + \lceil x/2 \rceil = k + k = 2k = x$

- When x is not divisible by 2 - i.e. $x = 2k + 1$, hence
  $\lfloor x/2 \rfloor = k$, $\lceil x/2 \rceil = k + 1$, and
  $\lfloor x/2 \rfloor + \lceil x/2 \rceil = k + k + 1 = 2k + 1 = x$

- 3.3.3/ The floor function $f(x) = \lfloor x \rfloor$ and the ceiling function $f(x) = \lceil x \rceil$ are monomotically increasing.
  Proof: we will show this property for the ceiling function, the proof for the floor function follows the same line of thoughts:
  Consider $x, y \in \mathbb{R}$:

$$y = \begin{cases} \lceil x \rceil & \text{if} \quad x \leq y \leq \lceil x \rceil \\ \lceil y \rceil \geq y > \lceil x \rceil \geq x & \text{if} \quad y > \lceil x \rceil \end{cases}$$

Hence if $x \leq y$, $\lceil x \rceil \leq \lceil y \rceil$ or $f(x) \leq f(y)$, hence QED.

- 3.3.4/ $\lceil a+b \rceil \leq \lceil a \rceil + \lceil b \rceil$ for $a, b \in \mathbb{R}$:
  Proof: We have: $\lceil a \rceil \geq a, \lceil b \rceil \geq b$
  Hence:

$$\lceil a \rceil + \lceil b \rceil = \lceil \lceil a \rceil + \lceil b \rceil \rceil \geq \lceil a+b \rceil$$

  The inequality occurs due to the monotonically increasing property of the ceiling function.

- 3.3.5/ $\lfloor a+b \rfloor \geq \lfloor a \rfloor + \lfloor b \rfloor$ for $a, b \in \mathbb{R}$:
  Proof: We have: $\lfloor a \rfloor \leq a, \lfloor b \rfloor \leq b$,
  Hence:

$$\lfloor a+b \rfloor \geq \lfloor \lfloor a \rfloor + \lfloor b \rfloor \rfloor = \lfloor a \rfloor + \lfloor b \rfloor$$

  The inequality occurs since the floor function is monotonically increasing.

- 3.3.6/ $\left\lceil \frac{\lceil x/a \rceil}{b} \right\rceil = \left\lceil \frac{x}{ab} \right\rceil$ for $x \in \mathbb{R}^+, a, b \in \mathbb{N}$

  Proof:
  Observe that the equality is true if x is divisible by a -i.e. $\lfloor x/a \rfloor = k \in \mathbb{N}$ or $b = 1$.
  Hence, we will show that the equality is still true if the remainder is not an integer. We have:

$$\lfloor x/a \rfloor = k < x/a < \lceil x/a \rceil = k+1$$

  Therefore, by the increasing property of ceiling function:

$$\lceil k/b \rceil \leq \lceil x/ab \rceil \leq \lceil (k+1)/b \rceil$$

  Consider when k is divisible by b, then

$$\lceil k/b \rceil < \lceil x/ab \rceil = \lceil (k+1)/b \rceil = \lceil k/b \rceil + 1$$

  The last equality occurs since $\lceil k/b \rceil + 1 = k/b + 1$ is the smallest integer upperbound for $\lceil (k+1)/b \rceil$.
  Otherwise, we have:

$$\lceil k/b \rceil = \lfloor k/b \rfloor + 1$$

  Let:

$$\frac{k+1}{b} = \left\lfloor \frac{k}{b} \right\rfloor + r$$

  It is easy to show that $r \in (0, 1]$: $0 < r = (k+1)/b - \lfloor k/b \rfloor \leq (k+1)/1 - k = 1$
  By the increasing property of the ceiling function:

$$\left\lceil \frac{k}{b} \right\rceil \leq \left\lceil \frac{k+1}{b} \right\rceil \leq \left\lfloor \frac{k}{b} \right\rfloor + \lceil r \rceil = \left\lfloor \frac{k}{b} \right\rfloor + 1 = \left\lceil \frac{k}{b} \right\rceil$$

Hence,

$$\left\lceil \frac{k}{b} \right\rceil = \left\lceil \frac{x}{ab} \right\rceil = \left\lceil \frac{k+1}{b} \right\rceil$$

Hence QED

- 3.3.7/ $\lfloor \lfloor x/a \rfloor /b \rfloor = \lfloor x/ab \rfloor$ for $x \in \mathbb{R}^+, a, b \in \mathbb{N}$:
  If x is divisible by a or $b = 1$, then the equality is fulfilled. Otherwise, let $k \in \mathbb{N} = \lfloor x/a \rfloor$, we have:

  $$k < x/a < k+1$$

  and

  $$\lfloor k/b \rfloor \leq \lfloor x/ab \rfloor \leq \lfloor (k+1)/b \rfloor$$

  Consider when k is divisible by b, then $\lfloor k/b \rfloor = k/b$, by the increasing property of the floor function:

  $$\lfloor k/b + 1/b \rfloor \leq k/b + 1$$

  The equality of which occurs only when b = 1, therefore, for $b > 1$:

  $$k/b = \lfloor k/b \rfloor = \lfloor x/ab \rfloor = \lfloor (k+1)/b \rfloor < k/b + 1$$

  When k is not divisible by b, again, let

  $$\frac{k+1}{b} = \left\lfloor \frac{k}{b} \right\rfloor + r$$

  Hence:

  $$\left\lfloor \frac{k+1}{b} \right\rfloor = \left\lfloor \frac{k}{b} \right\rfloor + \lfloor r \rfloor = \left\lfloor \frac{k}{b} \right\rfloor$$

  since $r = 1$ only when $b = 1$. Hence QED

- 3.3.8 Prove the following:

  $$\left\lceil \frac{a}{b} \right\rceil \leq \frac{a+b-1}{b}$$

  $$\left\lfloor \frac{a}{b} \right\rfloor \geq \frac{a-b+1}{b}$$

Proof:
When a is divisible by b, the inequality can easilly be shown. When a is not divisible by b, let:

$$\frac{m}{b} = \left\lceil \frac{a}{b} \right\rceil$$

$$\frac{n}{b} = \left\lfloor \frac{a}{b} \right\rfloor$$

Then,

$$\frac{m-n}{b}=1$$

Or

$$m-n=b$$

Hence

$$m-a<m-n=b$$

and

$$a-n<m-n=b$$

Or

$$m-a\leq b-1$$

and

$$a-n\leq b-1$$

Therefore,

$$\left\lceil\frac{a}{b}\right\rceil=\frac{m}{b}\leq\frac{a+b-1}{b}$$
$$\left\lfloor\frac{a}{b}\right\rfloor=\frac{n}{b}\geq\frac{a-b+1}{b}$$

QED

An alternate proof:
Consider the case that a is not divisible by b -i.e. $a=kb+r$ where $k,r\in\mathbb{N},1\leq r<b$.
Hence, the ceiling function:

$$\left\lceil\frac{a}{b}\right\rceil=\left\lceil\frac{kb+r}{b}\right\rceil=\frac{kb}{b}+1=\frac{kb}{b}+\frac{b-r+r}{b}=\frac{kb+r}{b}+\frac{b-r}{b}=\frac{a}{b}+\frac{b-r}{b}\leq\frac{a+(b-1)}{b}$$

The inequality occurs since $r\geq 1$.
The floor function:

$$\left\lfloor\frac{a}{b}\right\rfloor=\frac{kb}{b}=\frac{kb+r-r}{b}=\frac{a-r}{b}\geq\frac{a-(b-1)}{b}$$

The inequality occurs since $r<b,r\in\mathbb{N}$, or $r\leq b-1$

Polynomials:

Given a nonnegative integer d, a polynomial in n of degree d is:

$$p(n)=\sum_{i=0}^{d}a_in^i$$

We say that a function is polynomially bounded if $f(n) = O(n^k)$ for some constant k.

Exponentials:

Prove the following:

$$\lim_{n\to\infty} \frac{n^b}{a^n} = 0 \quad \text{for } a > 1$$

hence deduce that:

$$n^b = o(a^n)$$

Proof:
Use L'Hopital's rule - reproduced as follows:
Let f, g be differentiable on an open internal I, except possibly at $c \in I$, if $\lim_{n\to c} f(n) = \lim_{n\to c} g(n) = 0$ or ∞, and $g(n) \neq 0$ for all $n \in I, n \neq c$, and $\lim_{n\to c} \frac{f'(x)}{g'(x)}$ exists, then

$$\lim_{n\to c} \frac{f(n)}{g(n)} = \lim_{n\to c} \frac{f'(x)}{g'(x)}$$

Let $f(n) = n^b$ and $g(n) = a^n$. Note that $\lim_{n\to\infty} f(n) = \lim_{n\to\infty} g(n) = \infty$. We also have $f(n), g(n)$ are b times differentiable, and $g^{(m)}(n) \neq 0$ where m denotes the m order differentiation of $g(m)$, then we have:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{f'(x)}{g'(x)} = \cdots = \lim_{n\to\infty} \frac{f^{(b)}(n)}{g^{(b)}(n)} = \lim_{n\to\infty} \frac{b!}{a^n \log^b(a)} = 0$$

QED

Therefore, an exponential function with a base > 1 is strictly increasing and that it grows faster than any polynomial function.

Logarithms

Define a function $\log_b()$ such that $log_b(b^n) = n$ for $b \in \mathbb{R}$.
If $b > 1$ then $\log_b()$ is strictly increasing.
For all $a > 0, b > 0, c > 0, a, b, c \in \mathbb{R}, n \in \mathbb{N}$

- 3.3.9/ $a = b^{\log_b a}$
  Proof:

$$\begin{aligned} a &= b^{\log_b(a)} \\ \iff \log_b(a) &= \log_b(b^{\log_b(a)}) \quad \text{By definition.} \end{aligned}$$

- 3.3.10/ $\log_c(ab) = \log_c(a) + \log_c(b)$
  Proof:

$$\begin{aligned} & \log_c(ab) = \log_c(a) + \log_c(b) \\ \iff & ab = c^{\log_c(ab)} = c^{\log_c(a)+\log_c(b)} \\ \iff & ab = c^{\log_c(a)} \times c^{\log_c(b)} \quad \text{By definition} \end{aligned}$$

- 3.3.11/$\log_b(a^n) = n\log_b(a)$
  Proof:

$$\begin{aligned} & b^{\log_b a^n} = b^{n\log_b a} \\ \iff & a^n = (b^{\log_b a})^n \end{aligned}$$

- 3.3.12/ $\log_b(a) = \frac{1}{\log_a(b)}$
  Proof:

$$\begin{aligned} & \log_b a = \frac{1}{\log_a b} \\ \iff & \log_b a \times \log_a b = 1 \\ \iff & b^{\log_b a \times \log_a b} = b^1 = b \\ \iff & a^{\log_a b} = b \end{aligned}$$

- 3.3.13/ $\log_b(1/a) = -\log_b(a)$
  Proof:

$$\log_b(1/a) = \log_b(a^-1) = -1 \times \log_b(a) = -\log_b(a)$$

- 3.3.14/ $\log_b a = \log_c a/\log_c b$
  Proof:

$$\begin{aligned} & \log_b(a) = \frac{\log_c a}{\log_c b} \\ \iff & \log_b a \times \log_c b = \log_c a \\ \iff & c^{\log_b a \times \log_c b} = c^{\log_c a} = a \\ \iff & b^{\log_b a} = a \end{aligned}$$

Equation 3.3.14 allows us to change the base of logarithmic function from one constant to another by multiplying another constant, hence for the big O notation involving logarithmic terms, it does not matter the base and $\log$ is used implicitly with base e.
We say that a function is *polylogarithmically bounded* if $f(n) = O(\log^k n)$ where $\log^k n = (\log n)^k$ Note that any positive polynomial function grows faster than any polylogarithmic function. *Prove this*.

$$\log^k n = o(n^a) \quad \text{for } a > 0$$

Proof:
Again, make use of the L'Hopital rule, showing that :

$$\lim_{n\to\infty} \log^k n = \lim_{n\to\infty} n^a = \infty$$

with $f(n) = \log^k(n), g(n) = n^a$ being a times differentiable, hence

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{f^{(a)}(n)}{g^{(a)}(n)} = \lim_{n\to\infty} \frac{k!}{n^m} = 0$$

for some constant m.