# Chapter 5

# Asymptotic Notation

## 5.1 Run time and constant factors

When calculating the running time of a program, we may know how many basic "steps" it takes as a function of input size, but we may not know how long each step takes on a particular computer. We would like to estimate the overall running time of an algorithm while ignoring constant factors (like how fast the CPU is). So, for example, if we have 3 machines, where operations take $3\mu$s, $8\mu$s and $0.5\mu$s, the three functions measuring the amount of time required, $t(n) = 3n^2$, $t(n) = 8n^2$, and $t(n) = n^2/2$ are considered the same, ignoring ("to within") constant factors (the time required always grows according to a quadratic function in terms of the size of the input $n$).

The nice thing is that this means that lower order terms can be ignored as well! So $f(n) = 3n^2$ and $g(n) = 3n^2 + 2$ are considered "the same", as are $h(n) = 3n^2 + 2n$ and $j(n) = 5n^2$. Notice that

$$\forall n \in \mathbb{N}, n \geq 1 \Rightarrow f(n) \leq g(n) \leq h(n) \leq j(n)$$

but there's always a constant factor that can reverse any of these inequalities.

Really what we want to measure is the growth rate of functions (and in computer science, the growth rate of functions that bound the running time of algorithms). You might be familiar with binary search and linear search (two algorithms for searching for a value in a sorted array). Suppose one computer runs binary search and one computer runs linear search. Which computer will give an answer first, assuming the two computers run at roughly the same CPU speed? What if one computer is much faster (in terms of CPU speed) than the other, does it affect your answer? What if the array is really, really big?

## How large is "sufficiently large?"

Is binary search a better algorithm than linear search?[1] It depends on the size of the input. For example, suppose you established that linear search has complexity $L(n) = 3n$ and binary search has complexity $B(n) = 9 \log_2 n$. For the first few $n$, $L(n)$ is smaller than $B(n)$. However, certainly for $n > 10$, $B(n)$ is smaller, indicating less "work" for binary search.

When we say "large enough" $n$, we mean we are discussing the asymptotic behaviour of the complexity function, and we are prepared to ignore the behaviour near the origin.

## 5.2 Asymptotic notation: Making Big-O precise

We define $\mathbb{R}^{\geq 0}$ as the set of nonnegative real numbers, and define $\mathbb{R}^+$ as the set of positive real numbers. Here is a precise definition of "The set of functions that are eventually no more than $f$, to within a constant factor":

Definition: For any function $f : \mathbb{N} \to \mathbb{R}^{\geq 0}$ (*i.e.*, any function mapping naturals to nonnegative reals), let

$$\mathcal{O}(f) = \{g : \mathbb{N} \to \mathbb{R}^{\geq 0} \mid \exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow g(n) \leq cf(n)\}.$$

Saying $g \in \mathcal{O}(f)$ says that "$g$ grows no faster than $f$" (or equivalently, "$f$ is an upper bound for $g$"), so long as we modify our understanding of "growing no faster" and being an "upper bound" with the practice of ignoring constant factors. Now we can prove some theorems.

Suppose $g(n) = 3n^2 + 2$ and $f(n) = n^2$. Then $g \in \mathcal{O}(f)$. To be more precise, we need to prove the statement $\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow 3n^2 + 2 \leq cn^2$. It's enough to find some $c$ and $B$ that "work" in order to prove the theorem.

Finding $c$ means finding a factor that will scale $n^2$ up to the size of $3n^2 + 2$. Setting $c = 3$ almost works, but there's that annoying additional term 2. Certainly $3n^2 + 2 < 4n^2$ so long as $n \geq 2$, since $n \geq 2 \Rightarrow n^2 > 2$. So pick $c = 4$ and $B = 2$ (other values also work, but we like the ones we thought of first). Now concoct a proof of

$$\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow 3n^2 + 2 \leq cn^2.$$

Let $c' = 4$ and $B' = 2$.
Then $c' \in \mathbb{R}^+$ and $B' \in \mathbb{N}$.
Assume $n \in \mathbb{N}$ and $n \geq B'$. # direct proof for an arbitrary natural number
  Then $n^2 \geq B'^2 = 4$. # (squaring is monotonic on natural numbers.)
  Then $n^2 \geq 2$.
  Then $3n^2 + n^2 \geq 3n^2 + 2$. # (adding $3n^2$ to both sides of the inequality).
  Then $3n^2 + 2 \leq 4n^2 = c'n^2$ # re-write
Then $\forall n \in \mathbb{N}, n \geq B' \Rightarrow 3n^2 + 2 \leq c'n^2$ # introduce $\forall$ and $\Rightarrow$
Then $\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow 3n^2 + 2 \leq cn^2$. # introduce $\exists$ (twice)

So, by definition, $g \in \mathcal{O}(f)$.

### A more complex example

Let's prove that $2n^3 - 5n^4 + 7n^6$ is in $\mathcal{O}(n^2 - 4n^5 + 6n^8)$. We begin with:

Let $c' =$ ____. Then $c' \in \mathbb{R}^+$.
Let $B' =$ ____. Then $B' \in \mathbb{N}$.
Assume $n \in \mathbb{N}$ and $n \geq B'$. # arbitrary natural number and antecedent
  Then $2n^3 - 5n^4 + 7n^6 \leq \cdots \leq c'(n^2 - 4n^5 + 6n^8)$.
Then $\forall n \in \mathbb{N}, n \geq Bi' \Rightarrow 2n^3 - 5n^4 + 7n^6 \leq c'(n^2 - 4n^5 + 6n^8)$. # introduce $\Rightarrow$ and $\forall$
Hence, $\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow 2n^3 - 5n^4 + 7n^6 \leq c(n^2 - 4n^5 + 6n^8)$. # introduce $\exists$

To fill in the $\cdots$ we try to form a chain of inequalities, working from both ends, simplifying the expressions:

$$\begin{aligned}
2n^3 - 5n^4 + 7n^6 &\le 2n^3 + 7n^6 && \text{(drop } -5n^4 \text{ because it doesn't help us in an important way)}\\
&\le 2n^6 + 7n^6 && \text{(increase } n^3 \text{ to } n^6 \text{ because we have to handle } n^6 \text{ anyway)}\\
&= 9n^6 \\
&\le 9n^8 && \text{(simpler to compare)}\\
&= 2(9/2)n^8 && \text{(get as close to form of the simplified end result: now choose } c' = 9/2\text{)}\\
&= 2cn^8 \\
&= c'(-4n^8 + 6n^8) && \text{(reading bottom up: decrease } -4n^5 \text{ to } -4n^8 \text{ because we have to handle } n^8 \text{ anyway)}\\
&\le c'(-4n^5 + 6n^8) && \text{(reading bottom up: drop } n^2 \text{ because it doesn't help us in an important way)}\\
&\le c'(n^2 - 4n^5 + 6n^8)
\end{aligned}$$

We never needed to restrict $n$ in any way beyond $n \in \mathbb{N}$ (which includes $n \ge 0$), so we can fill in $c' = 9/2$, $B' = 0$, and complete the proof.

Let's use this approach to prove $n^4 \notin \mathcal{O}(3n^2)$. More precisely, we have to prove the negation of the statement $\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in N, n \ge B \Rightarrow n^4 \le c3n^2$.

Assume $c \in \mathbb{R}^+$ and $B \in \mathbb{N}$. # arbitrary positive real number and natural number
  Let $n_0 =$ ____.
  $\cdots$
  So $n_0 \in \mathbb{N}$.
  $\cdots$
  So $n_0 \ge B$.
  $\cdots$
  So $n_0^4 > c3n_0^2$.
Then $\forall c \in \mathbb{R}^+, \forall B \in \mathbb{N}, \exists n \in \mathbb{N}, n \ge B \wedge n^4 > c3n^2$.

Here's our chain of inequalities (the third $\cdots$):

$$\begin{aligned}
\text{And } n_0^4 &\ge n_0^3 && \text{(don't need full power of } n_0^4\text{)}\\
&= n_0 \cdot n_0^2 && \text{(make form as close as possible)}\\
&> c \cdot 3n_0^2 && \text{(if we make } n_0 > 3c \text{ and } n_0 > 0\text{).}
\end{aligned}$$

Now pick $n_0 = \max(B, \lceil 3c + 1 \rceil)$.
The first $\cdots$ is:

Since $c > 0$, $3c + 1 > 0$, so $\lceil 3c + 1 \rceil \in \mathbb{N}$.
Since $B \in \mathbb{N}$, $\max(B, \lceil 3c + 1 \rceil) \in \mathbb{N}$.

The second $\cdots$ is:

$\max(B, \lceil 3c + 1 \rceil) \ge B$.

We also note just before the chain of inequalities:

$n_0 = \max(B, \lceil 3c + 1 \rceil) \ge \lceil 3c + 1 \rceil \ge 3c + 1 > 3c$.

Some points to note are:

- Don't "solve" for $n$ until you've made the form of the two sides as close as possible.
- You're not exactly solving for $n$: you are finding a condition of the form $n >$ __ that makes the desired inequality true. You might find yourself using the "max" function a lot.

## Other bounds

In analogy with $\mathcal{O}(f)$, consider two other definitions:

Definition: For any function $f : \mathbb{N} \to \mathbb{R}^{\geq 0}$, let

$$\Omega(f) = \{g : \mathbb{N} \to \mathbb{R}^{\geq 0} \mid \exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow g(n) \geq cf(n)\}.$$

To say "$g \in \Omega(f)$" expresses the concept that "$g$ grows at least as fast as $f$" ($f$ is a lower bound on $g$).

Definition: For any function $f : \mathbb{N} \to \mathbb{R}^{\geq 0}$, let

$$\Theta(f) = \{g : \mathbb{N} \to \mathbb{R}^{\geq 0} \mid \exists c_1 \in \mathbb{R}^+, \exists c_2 \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow c_1 f(n) \leq g(n) \leq c_2 f(n)\}.$$

To say "$g \in \Theta(f)$" expresses the concept that "$g$ grows at the same rate as $f$" ($f$ is a tight bound for $g$, or $f$ is both an upper bound and a lower bound on $g$).

## Induction interlude

Suppose $P(n)$ is some predicate of the natural numbers, and:

$$(*) \quad P(0) \wedge (\forall n \in \mathbb{N}, P(n) \Rightarrow P(n+1)).$$

You should certainly be able to show that (*) implies $P(0)$, $P(1)$, $P(2)$, in fact $P(n)$ where $n$ is any natural number you have the patience to folow the chain of results to obtain. In fact, we feel that we can "turn the crank" enough times to show that (*) implies $P(n)$ for any natural number $n$. This is called the Principle of Simple Induction (PSI). It isn't proved, it is an axiom that we assume to be true.

Here's an an application of the PSI that will be useful for some big-Oh problems.

$P(n)$: $2^n \geq 2n$.

I'd like to prove that $\forall n, P(n)$, using the PSI. Here's what I do:

Prove $P(0)$: $P(0)$ states that $2^0 = 1 \geq 2(0) = 0$, which is true.

Prove $\forall n \in \mathbb{N}, P(n) \Rightarrow P(n+1)$:

Assume $n \in \mathbb{N}$. # arbitrary natural number
  Assume $P(n)$, that is $2^n \geq 2n$. # antecedent
  Then $n = 0 \vee n > 0$. # natural numbers are non-negative
  Case 1, assume $n = 0$: Then $2^{n+1} = 2^1 = 2 \geq 2(n+1) = 2$.
  Case 2, assume $n > 0$: Then $n \geq 1$. # $n$ is an integer greater than 0
    Then $2^n \geq 2$. # Since $n \geq 1$, and $2^n$ is monotone increasing
    Then $2^{n+1} = 2^n + 2^n \geq 2n + 2 = 2(n+1)$. # by previous line and IH $P(n)$
  Then $2^{n+1} \geq 2(n+1)$, which is $P(n+1)$. # true in both possible cases
 Then $P(n) \Rightarrow P(n+1)$. # introduce $\Rightarrow$

Then $\forall n \in \mathbb{N}, P(n) \Rightarrow P(n+1)$. # introduce $\forall$

I now conclude, by the PSI, $\forall n \in \mathbb{N}, P(n)$, that is $2^n \geq 2n$.

Here's a big-Oh problem where we can use $P(n)$. Let $g(n) = 2^n$ and $f(n) = n$. I want to show that $g \notin \mathcal{O}(f)$.

Assume $c \in \mathbb{R}^+$, assume $B \in \mathbb{N}$. # arbitrary values

Let $k = \lceil \log_2(c) \rceil + 1 + B$, and let $n_0 = 2k$.

Then $n_0 \in \mathbb{N}$. # $\lceil c \rceil, 1, 2, B \in \mathbb{N}$, $\mathbb{N}$ closed under $+, *$

Then $n_0 \geq B$. # At least twice $B$

Then $2^k > c$. # Choice of $k$, $2^x$ is increasing function.

Then

$$\begin{aligned} g(n_0) = 2^{n_0} &= 2^k \times 2^k & \text{\# by choice of } k \\ &\geq 2^k \times 2k & \text{\# by } P(2k) \\ &= 2^k \times n_0 > cn_0 & \text{\# since } n_0 = 2k \text{ and } 2^k > c \\ &= cf(n_0) \end{aligned}$$

Then $n_0 \geq B \wedge g(n_0) \geq cf(n_0)$. # introduce $\wedge$

Then $\exists n \in \mathbb{N}, n \geq B \wedge g(n) \geq cf(n)$. # introduce $\exists$

Then $\forall c \in \mathbb{R}, \forall B \in \mathbb{N}, \exists n \in \mathbb{N}, n \geq B \wedge g(n) > cf(n)$. # introduce $\forall$

So, I can conclude that $g \notin \mathcal{O}(f)$.

What happens to induction for predicates that are true for all natural numbers after a certain point, but untrue for the first few natural numbers? For example, $2^n$ grows much more quickly than $n^2$, but $2^3$ is not larger than $3^2$. Choose $n$ big enough, though, and it is true that:

$$P(n) : 2^n > n^2.$$

You can't prove this for all $n$, when it is false for $n = 2, n = 3$, and $n = 4$, so you'll need to restrict the domain and prove that for all natural numbers greater than 4, $P(n)$ is true. We don't have a slick way to restrict domains in our symbolic notation. Let's consider three ways to restrict the natural numbers to just those greater than 4, and then use induction.

RESTRICT BY SET DIFFERENCE: One way to restrict the domain is by set difference:

$$\forall n \in \mathbb{N} \setminus \{0, 1, 2, 3, 4\}, P(n)$$

Again, we'll need to prove $P(5)$, and then that $\forall n \in \mathbb{N} \setminus \{0, 1, 2, 3, 4\}, P(n) \Rightarrow P(n+1)$.

RESTRICT BY TRANSLATION: We can also restrict the domain by translating our predicate, so that $Q(n) = P(n+5)$, that is:

$$Q(n) : 2^{n+5} > (n+5)^2$$

Now our task is to prove $Q(0)$ is true (it is: $32 > 25$), and that for all $n \in \mathbb{N}$, $Q(n) \Rightarrow Q(n+1)$. This is simple induction.

RESTRICT USING IMPLICATION: Another method of restriction uses implication to restrict the domain where we claim $P(n)$ is true—in the same way as for sentences:

$$\forall n \in \mathbb{N}, n \geq 5 \Rightarrow P(n).$$

The expanded predicate $Q(n) : n \geq 5 \Rightarrow P(n)$ now fits our pattern for simple induction, and all we need to do is prove:

1. $Q(0)$ is true (it is vacuously true, since $0 \geq 5$ is false).
2. $\forall n \in \mathbb{N}, Q(n) \Rightarrow Q(n+1)$. This breaks into cases. If $n < 4$, then $Q(n)$ and $Q(n+1)$ are both vacuously true (the antecedents of the implication are false, since $n$ and $n+1$ are not greater

than, nor equal to, 5), so there is nothing to prove. If $n = 4$, then $Q(n)$ is vacuously true, but $Q(n+1)$ has a true antecedent ($5 \geq 5$), so we need to prove $Q(5)$ directly: $2^5 > 5^2$ is true, since $32 > 25$. For $n > 5$, we can depend on the assumption of the consequent of $Q(n-1)$ being true to prove $Q(n)$:

$$\begin{aligned} 2^n &= 2^{n-1} + 2^{n-1} \quad \# \text{ definition of } 2^n \\ &> 2(n-1)^2 \quad \# \text{ antecedent of } Q(n-1) \\ &= 2n^2 - 2n + 2 = n^2 + n(n-2) + 2 \geq n^2 + 2 > n^2 \quad \# \text{ since } n > 4 \geq 2 \end{aligned}$$

After all that work, it turns out that we need prove just two things:

1. $P(5)$
2. $\forall n \in \mathbb{N}$, If $n > 4$, then $P(n) \Rightarrow P(n+1)$.

This is the same as before, except now our base case is $P(5)$ rather than $P(0)$, and we get to use the fact that $n \geq 5$ in our induction step (if we need it).

Whichever argument you're comfortable with, notice that simple induction is basically the same: you prove the base case (which may now be greater than 0), and you prove the induction step.

## Some theorems

Here are some general results that we now have the tools to prove.

- $f \in \mathcal{O}(f)$.
- $(f \in \mathcal{O}(g) \wedge g \in \mathcal{O}(h)) \Rightarrow f \in \mathcal{O}(h)$.
- $g \in \Omega(f) \Leftrightarrow f \in \mathcal{O}(g)$.
- $g \in \Theta(f) \Leftrightarrow g \in \mathcal{O}(f) \wedge g \in \Omega(f)$.

Test your intuition about Big-O by doing the "scratch work" to answer the following questions:

- Are there functions $f, g$ such that $f \in \mathcal{O}(g)$ and $g \in \mathcal{O}(f)$ but $f \neq g$?[2]
- Are there functions $f, g$ such that $f \notin \mathcal{O}(g)$, and $g \notin \mathcal{O}(f)$?[3]

To show that $(f \in \mathcal{O}(g) \wedge g \in \mathcal{O}(h)) \Rightarrow f \in \mathcal{O}(h)$, we need to find a constant $c \in \mathbb{R}^+$ and a constant $B \in \mathbb{N}$, that satisfy:

$$\forall n \in \mathbb{N}, n \geq B \Rightarrow f(n) \leq ch(n).$$

Since we have constants that scale $h$ to $g$ and then $g$ to $f$, it seems clear that we need their product to scale $g$ to $f$. And if we take the maximum of the two starting points, we can't go wrong. Making this precise:

Theorem 1: For any functions $f, g, h : \mathbb{N} \to \mathbb{R}^{\geq 0}$, we have $(f \in \mathcal{O}(g) \wedge g \in \mathcal{O}(h)) \Rightarrow f \in \mathcal{O}(h)$.

Proof:

Assume $f \in \mathcal{O}(g) \wedge g \in \mathcal{O}(h)$.

So $f \in \mathcal{O}(g)$.
So $g \in \mathcal{O}(h)$.
So $\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n > B \Rightarrow f(n) \leq cg(n)$. # by defn. of $f \in \mathcal{O}(g)$)
Let $c_g \in \mathbb{R}^+, B_g \in \mathbb{N}$ be such that $\forall n \in \mathbb{N}, n \geq B \Rightarrow f(n) \leq c_g g(n)$.
So $\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow g(n) \leq ch(n)$. # by defn. of $g \in \mathcal{O}(h)$)
Let $c_h \in \mathbb{R}^+, B_h \in \mathbb{N}$ be such that $\forall n \in \mathbb{N}, n \geq B_h \Rightarrow g(n) \leq c_h h(n)$.
Let $c' = c_g c_h$. Let $B' = \max(B_g, B_h)$. Then, $c' \in \mathbb{R}^+$ (because $c_g, c_h \in \mathbb{R}^+$) and $B' \in \mathbb{N}$ (because $B_g, B_h \in \mathbb{N}$).
Assume $n \in \mathbb{N}$ and $n \geq B'$.
Then $n \geq B_h$ (by definition of max), so $g(n) \leq c_h h(n)$.
Then $n \geq B_g$ (by definition of max), so $f(n) \leq c_g g(n) \leq c_g c_h h(n)$.
So $f(n) \leq c' h(n)$.
Hence, $\forall n \in \mathbb{N}, n \geq B' \Rightarrow f(n) \leq c' h(n)$.
Therefore, $\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow f(n) \leq ch(n)$.
So $f \in \mathcal{O}(g)$, by definition.
So $(f \in \mathcal{O}(g) \wedge g \in \mathcal{O}(h)) \Rightarrow f \in \mathcal{O}(h)$.

To show that $g \in \Omega(f) \Leftrightarrow f \in \mathcal{O}(g)$, it is enough to note that the constant, $c$, for one direction is positive, so its reciprocal will work for the other direction.[4]

THEOREM 2: For any functions $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$, we have $g \in \Omega(f) \Leftrightarrow f \in \mathcal{O}(g)$.

PROOF:
$g \in \Omega(f)$
$\Leftrightarrow$ (by definition)
$\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow g(n) \geq cf(n)$
$\Leftrightarrow$ (by letting $c' = 1/c$ and $B' = B$)
$\exists c' \in \mathbb{R}^+, \exists B' \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B' \Rightarrow f(n) \leq c' g(n)$
$\Leftrightarrow$ (by definition)
$f \in \mathcal{O}(g)$

To show $g \in \Theta(f) \Leftrightarrow g \in \mathcal{O}(f) \wedge g \in \Omega(f)$, it's really just a matter of unwrapping the definitions.

THEOREM 3: For any functions $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$, we have $g \in \Theta(f) \Leftrightarrow g \in \mathcal{O}(f) \wedge g \in \Omega(f)$.

PROOF:
$g \in \Theta(f)$
$\Leftrightarrow$ (by definition)
$\exists c_1 \in \mathbb{R}^+, \exists c_2 \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow c_1 f(n) \leq g(n) \leq c_2 f(n)$.
$\Leftrightarrow$ (combined inequality, and $B = \max(B_1, B_2)$)
$\left(\exists c_1 \in \mathbb{R}^+, \exists B_1 \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B_1 \Rightarrow g(n) \geq c_1 f(n)\right) \wedge$
$\left(\exists c_2 \in \mathbb{R}^+, \exists B_2 \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B_2 \Rightarrow g(n) \leq c_2 f(n)\right)$
$\Leftrightarrow$ (by definition)
$g \in \Omega(f) \wedge g \in \mathcal{O}(f)$

## TAXONOMY OF RESULTS

A LEMMA is a small result needed to prove something we really care about. A THEOREM is the main result that we care about (at the moment). A COROLLARY is an easy (or said to be easy) consequence of another result. A CONJECTURE is something suspected to be true, but not yet proven.

Here's an example of a conjecture whose proof has evaded the best minds for over 70 years. Maybe you'll prove it.

Define $f(n)$, for $n \in \mathbb{N}$ by:

$$f(n) = \begin{cases} n/2, & n \text{ even} \\ 3n+1, & n \text{ odd} \end{cases}$$

Let's define $f^2(n)$ as $f(f(n))$, and more generally, $f^{k+1}(n)$ as $f(f^k(n))$ for all $k \in \mathbb{N}$ (with the special case $f^0(n) = n$). Here's the conjecture:

CONJECTURE: $\forall n \in \mathbb{N}, n \geq 1 \Rightarrow \exists k \in \mathbb{N}, f^k(n) = 1$.

Easy to state, but (so far) hard to prove or disprove.

Here's an example of a corollary that recycles some of the theorems we've already proven (so we don't have to do the grubby work). To show $g \in \Theta(f) \Leftrightarrow f \in \Theta(g)$, I re-use theorems proved above and the commutativity of $\wedge$:

COROLLARY: For any functions $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$, we have $g \in \Theta(f) \Leftrightarrow f \in \Theta(g)$.

PROOF:

$g \in \Theta(f)$
$\Leftrightarrow$ (by Theorem 3)
$g \in \mathcal{O}(f) \wedge g \in \Omega(f)$.
$\Leftrightarrow$ (by Theorem 2)
$g \in \mathcal{O}(f) \wedge f \in \mathcal{O}(g)$
$\Leftrightarrow$ (by commutativity of $\wedge$)
$f \in \mathcal{O}(g) \wedge g \in \mathcal{O}(f)$
$\Leftrightarrow$ (by Theorem 2)
$f \in \mathcal{O}(g) \wedge f \in \Omega(g)$
$\Leftrightarrow$ (by Theorem 3)
$f \in \Theta(g)$.

## A VERY IMPORTANT NOTE

Note that asymptotic notation (the Big-O, Big-$\Omega$, and Big-$\Theta$ definitions) bound the asymptotic growth rates of *functions*, as $n$ approaches infinity. Often in computer science we use this asymptotic notation to bound functions that express the running times of algorithms, perhaps in best case or in worst case. Asymptotic notation *does not* express or bound the worst case or best case running time, only the functions expressing these values.

This distinction is subtle, but crucial to understanding both running times and asymptotic notation. If this warning doesn't seem important to you now, come back and read this again in a few weeks, months, or courses.

## EXERCISES

1. Prove or disprove the following claims:
   (a) $7n^3 + 11n^2 + n \in \mathcal{O}(n^3)$ [5]
   (b) $n^2 + 165 \in \Omega(n^4)$
   (c) $n! \in \mathcal{O}(n^n)$

(d) $n \in \mathcal{O}(n \log_2 n)$

(e) $\forall k \in \mathbb{N}, k > 1 \Rightarrow \log_k n \in \Theta(\log_2 n)$

2. Define $g(n) = \begin{cases} n^3/165, & n < 165 \\ \left\lceil \sqrt{6n^5} \right\rceil, & n \geq 165 \end{cases}$. Note that $\forall x \in \mathbb{R}, x \leq \lceil x \rceil < x + 1$.

   Prove that $g \in \mathcal{O}(n^{2.5})$.

3. Let $\mathcal{F}$ be the set of functions from $\mathbb{N}$ to $\mathbb{R}^{\geq 0}$. Prove the following theorems:

   (a) For $f, g \in \mathcal{F}$, if $g \in \Omega(f)$ then $g^2 \in \Omega(f^2)$.

   (b) $\forall k \in \mathbb{N}, k > 1 \Rightarrow \forall d \in \mathbb{R}^+, d \log_k n \in \Theta(\log_2 n)$. [6]

   Notice that (b) means that all logarithms eventually grow at the same rate (up to a multiplicative constant), so the base doesn't matter (and can be omitted inside the asymptotic notation).

4. Let $\mathcal{F}$ be the set of functions from $\mathbb{N}$ to $\mathbb{R}^{\geq 0}$. Prove or disprove the following claims:

   (a) $\forall f \in \mathcal{F}, \forall g \in \mathcal{F}, f \in \mathcal{O}(g) \Rightarrow (f + g) \in \Theta(g)$

   (b) $\forall f \in \mathcal{F}, \forall f' \in \mathcal{F}, \forall g \in \mathcal{F}, (f \in \mathcal{O}(g) \wedge f' \in \mathcal{O}(g)) \Rightarrow (f + f') \in \mathcal{O}(g)$

5. For each function $f$ in the left column, choose one expression $\mathcal{O}(g)$ from the right column such that $f \in \mathcal{O}(g)$. Use each expression exactly once.

| | |
|---|---|
| (i) $3 \cdot 2^n \in$ ________ | (a) $\mathcal{O}(\frac{1}{n})$ |
| (ii) $\frac{2n^4+1}{n^3+2n-1} \in$ ________ | (b) $\mathcal{O}(1)$ |
| (iii) $(n^5 + 7)(n^5 - 7) \in$ ________ | (c) $\mathcal{O}(\log_2 n)$ |
| (iv) $\frac{n^4 - n \log_2 n}{n^2+1} \in$ ________ | (d) $\mathcal{O}(n)$ |
| (v) $\frac{n \log_2 n}{n-5} \in$ ________ | (e) $\mathcal{O}(n \log_2 n)$ |
| (vi) $8 + \frac{1}{n^2} \in$ ________ | (f) $\mathcal{O}(n^2)$ |
| (vii) $2^{3n+1} \in$ ________ | (g) $\mathcal{O}(n^{10})$ |
| (viii) $n! \in$ ________ | (h) $\mathcal{O}(2^n)$ |
| (ix) $\frac{5 \log_2(n+1)}{1 + n \log_2 3n} \in$ ________ | (i) $\mathcal{O}(10^n)$ |
| (x) $(n-2)\log_2(n^3+4) \in$ ________ | (j) $\mathcal{O}(n^n)$ |

# Chapter 5 Notes

[1]Better in the sense of time complexity.

[2]Sure, $f = n^2$, $g = 3n^2 + 2$.

[3]Sure. $f$ and $g$ don't need to both be monotonic, so let $f(n) = n^2$ and

$$g(n) = \begin{cases} n, & n \text{ even} \\ n^3, & n \text{ odd} \end{cases}$$

So not every pair of functions from $\mathbb{N} \to \mathbb{R}^{\geq 0}$ can be compared using Big-O.

[4]Let's try the symmetrical presentation of bi-implication.

[5]The claim is true.
Let $c' = 8$. Then $c' \in \mathbb{R}^+$.
Let $B' = 12$. Then $B' \in \mathbb{N}$.
Assume $n \in \mathbb{N}$ and $n \geq B'$.
Then $n^3 = n \cdot n^2 \geq 12 \cdot n^2 = 11n^2 + n^2 \geq 11n^2 + n$. # since $n \geq B' = 12$
Thus $c'n^3 = 8n^3 = 7n^3 + n^3 \geq 7n^3 + 11n^2 + n$.
So $\forall n \in \mathbb{N}$, $n \geq B' \Rightarrow 7n^3 + 11n^2 + n \leq c'n^3$.
Since $B'$ is a natural number, $\exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow 7n^3 + 11n^2 + n \leq c'n^3$.
Since $c'$ is a real positive number, $\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow 7n^3 + 11n^2 + n \leq cn^3$.
By definition, $7n^3 + 11n^2 + n \in \mathcal{O}(n^3)$.

[6]

Assume $k \in \mathbb{N}$ and $k > 1$.
Assume $d \in \mathbb{R}^+$.
It suffices to argue that $d \log_k n \in \Theta(\log_2 n)$.
Let $c_1' = \frac{d}{\log_2 k}$. Since $k > 1$, $\log_2 k \neq 0$ and so $c_1' \in \mathbb{R}^+$.
Let $c_2' = \frac{d}{\log_2 k}$. By the same reasoning, $c_2' \in \mathbb{R}^+$.
Let $B' = 1$. Then $B' \in \mathbb{N}$.
Assume $n \in \mathbb{N}$ and $n \geq B'$.
Then $c_1' \log_2 n = \frac{d}{\log_2 k} \log_2 n = d\frac{\log_2 n}{\log_2 k} = d \log_k n \leq d \log_k n$.
Moreover, $d \log_k n \leq d\frac{\log_2 n}{\log_2 k} = \frac{d}{\log_2 k} \log_2 n = c_2' \log_2 n$.
Hence, $\forall n \in \mathbb{N}, n \geq B' \Rightarrow c_1' \log_2 n \leq d \log_k n \leq c_2' \log_2 n$.
Thus $\exists c_1 \in \mathbb{R}^+, \exists c_2 \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow c_1 \log_2 n \leq d \log_k n \leq c_2 \log_2 n$.
By definition, $d \log_k n \in \Theta(\log_2 n)$.
Thus, $\forall d \in \mathbb{R}^+, d \log_k n \in \Theta(\log_2 n)$.
Hence $\forall k \in \mathbb{N}, k > 1 \Rightarrow \forall d \in \mathbb{R}^+, d \log_k n \in \Theta(\log_2 n)$.