Section 3.1 – Asymptotic notation

3.1-1 Let $f(n)$ and $g(n)$ be asymptotically nonnegative functions. Using the basic definition of Θ-notation, prove that $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

Since $f(n)$ and $g(n)$ are both asymptotically nonnegative,

$$\exists\, n_0 \mid f(n) \geq 0 \; g(n) \geq 0 \;\; \forall\, n \geq n_0.$$

From the definition of $\Theta(\cdot)$, we have

$$\exists\, c_1 \; c_2 \; n_0 \in \mathbb{R}^+ \mid c_1 f(n) + c_1 g(n) \leq \max(f(n), g(n)) \leq c_2 f(n) + c_2 g(n) \;\; \forall\, n \geq n_0.$$

If $f(n) \geq g(n)$, we have

$$c_1 f(n) + c_1 g(n) \leq f(n) \leq c_2 f(n) + c_2 g(n).$$

The right-hand-side inequality is trivially satisfied with $c_2 = 1$. To find c_1, we notice that,

$$f(n) + g(n) \leq 2f(n),$$

and say,

$$c_1 = \frac{1}{2}.$$

The demonstration is similar for $g(n) > f(n)$, with $c_1 = 1/2$ and $c_2 = 1$.

3.1-2 Show that for any real constants a and b, where $b > 0$, $(n + a)^b = \Theta(n^b)$.

From the definition of $\Theta(\cdot)$, we have

$$\exists\, c_1 \; c_2 \; n_0 \in \mathbb{R}^+ \mid c_1 n^b \leq (n + a)^b \leq c_2 n^b \;\; \forall\, n \geq n_0,$$

and from the binomial theorem, we have

$$(n + a)^b = \binom{b}{0} n^b a^0 + \binom{b}{1} n^{b-1} a^1 + \cdots + \binom{b}{b-1} n^1 a^{b-1} + \binom{b}{b} n^0 a^b.$$

To find c_1, we notice that for n big enough,

$$\binom{b}{i} n^{b-i} a^i + \binom{b}{i+1} n^{b-(i+1)} a^{i+1} \geq 0 \quad \forall\, i \in 0, 2, \ldots, b,$$

which implies

$$\binom{b}{0} n^b a^0 + \binom{b}{1} n^{b-1} a^1 \leq (n + a)^b,$$

and also for n big enough,

$$\frac{n^b}{2} \leq n^b + \binom{b}{1} n^{b-1} a^1,$$

which implies

$$\frac{n^b}{2} \leq (n + a)^b,$$

and say

$$c_2 = \frac{1}{2}.$$

To find c_2, we notice that for n big enough,

$$n^b = \binom{b}{0} n^b a^0 \geq \binom{b}{i} n^{b-i} a^i \quad \forall\, i \in 1, \ldots, b,$$

which implies

$$(n + a)^b \leq (b + 1) n^b,$$

and say

$$c_2 = b + 1.$$

3.1-3 Explain why the statement, “The running time of algorithm A is at least $O(n^2)$,” is meaningless.

Because the O-notation only bounds from the top, not from the bottom.

3.1-4 Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$?

From the definition of $O(\cdot)$, we have

$$\exists\, c\; n_0 \in \mathbb{R}^+ \mid 0 \leq 2^{n+1} \leq c \cdot 2^n \;\; \forall\, n \geq n_0.$$

To find c, we notice that,

$$2^{n+1} = 2 \cdot 2^n,$$

and say $c = 2$ and $n_0 = 0$.
From the definition of $O(\cdot)$, we have

$$\exists\, c\; n_0 \in \mathbb{R}^+ \mid 0 \leq 2^{2n} \leq c \cdot 2^n \;\; \forall\, n \geq n_0.$$

To show that $2^{2n} \neq O(2^n)$, we notice that,

$$2^{2n} = 2^n \cdot 2^n,$$

which implies

$$c \geq 2^n,$$

which is not possible, since c is a constant and n is not.

3.1-5 Prove Theorem 3.1.

To prove

$$f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) \wedge f(n) = \Omega(g(n)).$$

we need to show

$$f(n) = O(g(n)) \wedge f(n) = \Omega(g(n)) \rightarrow f(n) = \Theta(g(n)),$$

and

$$f(n) = \Theta(g(n)) \rightarrow f(n) = O(g(n)) \wedge f(n) = \Omega(g(n)).$$

From the definition of $O(\cdot)$, we have

$$\exists\, c_1\; n_1 \in \mathbb{R}^+ \mid 0 \leq f(n) \leq c_1 g(n) \;\; \forall\, n \geq n_1,$$

and from the definition of $\Omega(\cdot)$, we have

$$\exists\, c_2\; n_2 \in \mathbb{R}^+ \mid 0 \leq c_2 g(n) \leq f(n) \;\; \forall\, n \geq n_2,$$

which implies

$$\exists\, c_1\; c_2 \in \mathbb{R}^+\; n_0 = \max(n_1, n_2) \mid c_2 g(n) \leq f(n) \leq c_1 g(n) \;\; \forall\, n \geq n_0 \iff f(n) = \Theta(g(n)).$$

From the definition of $\Theta(\cdot)$, we have

$$\exists\, c_1\; c_2\; n_0 \in \mathbb{R}^+ \mid c_2 g(n) \leq f(n) \leq c_1 g(n) \;\; \forall\, n \geq n_0,$$

which implies

$$\exists\, c_1\; n_0 \in \mathbb{R}^+ \mid 0 \leq f(n) \leq c_1 g(n) \;\; \forall\, n \geq n_0 \iff f(n) = O(g(n)),$$

$$\exists\, c_2\; n_0 \in \mathbb{R}^+ \mid c_2 g(n) \leq f(n) \leq 0 \;\; \forall\, n \geq n_0 \iff f(n) = \Omega(g(n)).$$

3.1-6 Prove that the running time of an algorithm is $\Theta(g(n))$ if and only if its worst-case running time is $O(g(n))$ and its best-case running time is $\Omega(g(n))$.

Let $f_b(n)$ and $f_w(n)$ be the best and worst-case running times of algorithm A, respectivelly.
If the running time of A is $\Theta(g(n))$, we have

$$f_b(n) = \Theta(g(n)),$$

and

$$f_w(n) = \Theta(g(n)).$$

From Theorem 3.1,

$$f_b(n) = \Theta(g(n)) \iff f_b(n) = O(g(n)) \wedge f_b(n) = \Omega(g(n)),$$

and

$$f_w(n) = \Theta(g(n)) \iff f_w(n) = O(g(n)) \wedge f_w(n) = \Omega(g(n)).$$

3.1-7 Prove that $o(g(n)) \cap \omega(g(n))$ is the empty set.

From the definition of $o(\cdot)$, we have

$$o(g(n)) = \{f(n) : \forall\, c_1 > 0\ \exists\, n_1 \in \mathbb{R}^+ \mid 0 \leq f(n) < c_1 g(n)\ \ \forall\, n \geq n_1\},$$

and from the definition of $\omega(\cdot)$, we have

$$\omega(g(n)) = \{f(n) : \forall\, c_2 > 0\ \exists\, n_2 \in \mathbb{R}^+ \mid 0 \leq c_2 g(n) < f(n)\ \ \forall\, n \geq n_2\}.$$

Thus,

$$o(g(n)) \cap \omega(g(n)) = \{f(n) : \forall\, c_1 > 0\ \forall\, c_2 > 0\ \exists\, n_0 \in \mathbb{R}^+ \mid 0 \leq c_2 g(n) < f(n) < c_1 g(n)\ \ \forall\, n \geq n_2\},$$

which is the empty set since, for very large n, $f(n)$ cannot be less than $c_1 g(n)$ and greater than $c_2 g(n)$ for all $c_1, c_2 > 0$.

3.1-8 We can extend our notation to the case of two parameters n and m that can go to infinity independently at different rates. For a given $g(n, m)$, we denote by $O(g(n, m))$ the set of functions

$$O(g(n, m)) = \{f(n, m) : \text{there exist positive constants } c, n_0, \text{and } m_0 \text{ such that } 0 \leq f(n, m) \leq cg(n, m) \text{ for all } n \geq n_0 \text{ and } m \geq m_0\}.$$

Give corresponding definitions for $\Omega(g(n, m))$ and $\Theta(g(n, m))$.

We denote by $\Omega(g(n, m))$ the set of functions

$$\Omega(g(n, m)) = \{f(n, m) : \exists\, c\ n_0\ m_0 \in \mathbb{R}^+ \mid 0 \leq cg(n, m)) \leq f(n, m)\ \ \forall\, n \geq n_0\ \ \forall\, m \geq m_0\}.$$

We denote by $\Theta(g(n, m))$ the set of functions

$$\Theta(g(n, m)) = \{f(n, m) : \exists\, c_1\ c_2\ n_0\ m_0 \in \mathbb{R}^+ \mid 0 \leq c_1 g(n, m) \leq f(n, m) \leq c_2 g(n, m)\ \ \forall\, n \geq n_0\ \ \forall\, m \geq m_0\}.$$

Section 3.2 – Standard notations and common functions

3.2-1 Show that if $f(n)$ and $g(n)$ are monotonically increasing functions, then so are the functions $f(n)+g(n)$ and $f(g(n))$, and if $f(n)$ and $g(n)$ are in addition nonnegative, then $f(n) \cdot g(n)$ is monotonically increasing.

If $f(n)$ and $g(n)$ are both monitonically increasing and $n \leq m$, we have

$$f(n) \leq f(m) \quad \text{and} \quad g(n) \leq g(m),$$

which implies that

$$f(n) - f(m) \leq 0 \quad \text{and} \quad g(n) - g(m) \leq 0.$$

Adding the above inequalities together, we have

$$f(n) - f(m) + g(n) - g(m) \leq 0 \to f(n) + g(n) \leq f(m) + g(m),$$

which shows that $f(n) + g(n)$ is monitonically increasing.
Also, let $g(n) = p$ and $g(m) = q$. Since $f(n) \leq f(m)$ and $g(n) \leq g(m)$, we have

$$f(p) \leq f(q) \to f(g(n)) \leq f(g(m)),$$

which shows that $f(g(n))$ is monitonically increasing.
If in addition, $f(\cdot) \geq 0$ and $g(\cdot) \geq 0$, we have

$$f(n) \leq f(m) \to f(n)g(n) \leq f(m)g(n) \to f(n)g(n) \leq f(m)g(m),$$

which shows that $f(n) \cdot g(n)$ is monitonically increasing.

3.2-2 Prove equation (3.16).

For all real $a > 0, b > 0, c > 0$,

$$a^{\log_b c} = a^{\frac{\log_a c}{\log_a b}} = \left(a^{\log_a c}\right)^{\frac{1}{\log_a b}} = c^{\frac{1}{\log_a b}} = c^{\log_b a}.$$

3.2-3 Prove equation (3.19). Also prove that $n! = \omega(2^n)$ and $n! = o(n^n)$.

Using the Stirling's approximation, we have

$$\begin{aligned}
\lg(n!) &\approx \lg\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1+\Theta\left(\frac{1}{n}\right)\right)\right) \\
&= \lg\left(\sqrt{2\pi n}\right) + \lg(\sqrt{n}) + \lg(n^n) - \lg(e^n) + \Theta(\lg(1/n)) \\
&= \Theta(1) + 1/2\lg(n) + n\lg n - n\lg e + \Theta(\lg(1/n)) \\
&= \Theta(1) + \Theta(\lg n) + \Theta(n\lg n) - \Theta(n) + \Theta(\lg(1/n)) \\
&= \Theta(n\lg n),
\end{aligned}$$

which proves Equation (3.19).
We have

$$n! = n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1 < \underbrace{n \cdot n \cdot n \cdots}_{\text{n times}} = n^n \;\; \forall\, n \geq 2,$$

which implies

$$n! = o(n^n).$$

We have

$$n! = n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1 > \underbrace{2 \cdot 2 \cdot 2 \cdots}_{\text{n times}} = 2^n \;\; \forall\, n \geq 4,$$

which implies

$$n! = w(2^n).$$

3.2-4 ($\star$) Is the function $\lceil \lg n \rceil!$ polynomially bounded? Is the function $\lceil \lg \lg n \rceil!$ polynomially bounded?

A function $f(n)$ is polynomially bounded if there are constants c, k, n_0 such that for all $n \geq n_0$, $f(n) \leq cn^k$. Thus, $\lg(f(n)) \leq ck \lg n$.
We have

$$\lg(\lceil \lg n \rceil!) = \Theta(\lceil \lg n \rceil \lg(\lceil \lg n \rceil)) = \Theta(\lg n \lg \lg n) = w(\lg n),$$

which implies that $\lg(\lceil \lg n \rceil!) > ck \lg n$, i.e., $\lceil \lg n \rceil!$ is not polynomially bounded.
We have

$$\lg(\lceil \lg \lg n \rceil!) = \Theta(\lceil \lg \lg n \rceil \lg \lceil \lg \lg n \rceil) = \Theta(\lg \lg n \lg \lg \lg n) = o(\lg^2 \lg n) = o(\lg^2 n) = o(\lg n),$$

which implies that $\lg(\lceil \lg \lg n \rceil!) \leq ck \lg n, i.e.,$ $\lceil \lg \lg n \rceil!$ is polynomially bounded.

3.2-5 ($\star$) Which is asymptotically larger: $\lg(\lg^\star n)$ or $\lg^\star(\lg n)$?

Let's assume that $\lg^*(x) = k$.
We have

$$\lg(\lg^* x) = \lg k,$$

and

$$\lg^*(\lg x) = k - 1,$$

since the inner logarithm that is applied to x will reduce the number of iterations of the iterative logarithm by 1.
Thus, since $(k-1)$ is asymptotically larger than $\lg(k)$, $\lg^*(\lg x)$ is also asymptotically larger than $\lg(\lg^* x)$.

3.2-6 Show that the golden ration ϕ and its conjugate $\hat{\phi}$ both satisfy the equation $x^2 = x + 1$.

The demonstration follows directly from the formulas of ϕ and $\hat{\phi}$.

$$\phi^2 = \left(\frac{1+\sqrt{5}}{2}\right)^2 = \frac{1+2\sqrt{5}+5}{4} = \frac{2\sqrt{5}+6}{4} = \frac{\sqrt{5}+3}{2} = \frac{1+\sqrt{5}}{2} + 1 = \phi + 1.$$

$$\hat{\phi}^2 = \left(\frac{1-\sqrt{5}}{2}\right)^2 = \frac{1-2\sqrt{5}+5}{4} = \frac{6-2\sqrt{5}}{4} = \frac{3-\sqrt{5}}{2} = \frac{1-\sqrt{5}}{2} + 1 = \hat{\phi} + 1.$$

3.2-7 Prove by induction that the ith Fibonacci number satisfies the equality

$$F_i = \frac{\phi^i - \hat{\phi}^i}{\sqrt{5}},$$

where ϕ is the golden ratio and $\hat{\phi}$ is its conjugate.

We have that

$$F_0 = \frac{\phi^0 - \hat{\phi}^0}{\sqrt{5}} = \frac{1-1}{\sqrt{5}} = 0,$$

and

$$F_1 = \frac{\phi^1 - \hat{\phi}^1}{\sqrt{5}} = \frac{1+\sqrt{5}-1+\sqrt{5}}{2\sqrt{5}} = \frac{2\sqrt{5}}{2\sqrt{5}} = 1.$$

which are the correct Fibonacci values for $i = 0$ and $i = 1$. Then we have the inductive step:

$$\begin{aligned}
F_i + F_{i+1} &= \frac{\phi^i + \hat{\phi}^i}{\sqrt{5}} + \frac{\phi^{i+1} + \hat{\phi}^{i+1}}{\sqrt{5}} \\
&= \frac{\phi^i + \phi^{i+1} - (\hat{\phi}^i + \hat{\phi}^{i+1})}{\sqrt{5}} \\
&= \frac{\phi^i(1+\phi) - \hat{\phi}^i(1+\phi)}{\sqrt{5}} \\
&= \frac{\phi^i\phi^2 - \hat{\phi}^i\hat{\phi}^2}{\sqrt{5}} \\
&= \frac{\phi^{i+2} - \hat{\phi}^{i+2}}{\sqrt{5}} = F_{i+2}.
\end{aligned}$$

3.2-8 Show that $k \ln k = \Theta(n)$ implies $k = \Theta(n/\ln n)$.

From the symmetry of Θ, we have

$$k \ln k = \Theta(n) \to n = \Theta(k \ln k),$$

and

$$\ln n = \Theta(\ln(k \ln k)) = \Theta(\ln k \ln \ln k) = \Theta(\ln k).$$

Thus,

$$\frac{n}{\ln n} = \frac{\Theta(k \ln k)}{\Theta(\ln k \ln \ln k)} = \Theta\left(\frac{k \ln k}{\ln k \ln \ln k}\right) = \Theta(k),$$

which implies

$$k = \Theta\left(\frac{n}{\ln n}\right).$$

Problems

Skipped for later.