3 Growth of Functions

3.1 Asymptotic notation

3.1-1

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

For asymptotically nonnegative functions $f(n)$ and $g(n)$, we know that

$$\begin{aligned} \exists n_1, n_2: f(n) &\ge 0 & \text{for, } n > n_1 \\ g(n) &\ge 0 & \text{for, } n > n_2. \end{aligned}$$

Let $n_0 = \max(n_1, n_2)$ and we know the equations below would be true for $n > n_0$:

$$\begin{aligned} f(n) &\le \max(f(n), g(n)) \\ g(n) &\le \max(f(n), g(n)) \\ (f(n) + g(n))/2 &\le \max(f(n), g(n)) \\ \max(f(n), g(n)) &\le (f(n) + g(n)). \end{aligned}$$

Then we can combine last two inequalities:

$$0 \le \frac{f(n) + g(n)}{2} \le \max(f(n), g(n)) \le f(n) + g(n).$$

Which is the definition of $\Theta(f(n) + g(n))$ with $c_1 = \frac{1}{2}$ and $c_2 = 1$

3.1-2

> Show that for any real constants a and b, where $b > 0$,
>
> $$(n + a)^b = \Theta(n^b). \tag{3.2}$$

Expand $(n + a)^b$ by the Binomial Expansion, we have

$$(n + a)^b = C_0^b n^b a^0 + C_1^b n^{b-1} a^1 + \cdots + C_b^b n^0 a^b.$$

Besides, we know below is true for any polynomial when $x \ge 1$.

$$a_0 x^0 + a_1 x^1 + \cdots + a_n x^n \le (a_0 + a_1 + \cdots + a_n) x^n.$$

Thus,

$$C_0^b n^b \le C_0^b n^b a^0 + C_1^b n^{b-1} a^1 + \cdots + C_b^b n^0 a^b \le (C_0^b + C_1^b + \cdots + C_b^b) n^b = 2^b n^b.$$

$$\implies (n + a)^b = \Theta(n^b).$$

3.1-3

> Explain why the statement, "The running time of algorithm A is at least $O(n^2)$," is meaningless.

$T(n)$: running time of algorithm A. We just care about the upper bound and the lower bound of $T(n)$.

The statement: $T(n)$ is at least $O(n^2)$.

- Upper bound: Because "$T(n)$ is at least $O(n^2)$", there's no information about the upper bound of $T(n)$.
- Lower bound: Assume $f(n) = O(n^2)$, then the statement: $T(n) \geq f(n)$, but $f(n)$ could be any fuction that is "smaller" than n^2. For example, constant, n, etc, so there's no conclusion about the lower bound of $T(n)$, too.

Therefore, the statement, "The running time of algorithm A is at least $O(n^2)$," is meaningless.

3.1-4

> Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$?

- True. Note that $2^{n+1} = 2 \times 2^n$. We can choose $c \geq 2$ and $n_0 = 0$, such that $0 \leq 2^{n+1} \leq c \times 2^n$ for all $n \geq n_0$. By definition, $2^{n+1} = O(2^n)$.

- False. Note that $2^{2n} = 2^n \times 2^n = 4^n$. We can't find any c and n_0, such that $0 \leq 2^{2n} = 4^n \leq c \times 2^n$ for all $n \geq n_0$.

3.1-5

> Prove Theorem 3.1.

The theorem states:

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

From $f = \Theta(g(n))$, we have that

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \text{ for } n > n_0.$$

We can pick the constants from here and use them in the definitions of O and Ω to show that both hold.

From $f(n) = \Omega(g(n))$ and $f(n) = O(g(n))$, we have that

$$\begin{aligned} & 0 \leq c_3 g(n) \leq f(n) & & \text{for all } n \geq n_1 \\ \text{and } & 0 \leq f(n) \leq c_4 g(n) & & \text{for all } n \geq n_2. \end{aligned}$$

If we let $n_3 = \max(n_1, n_2)$ and merge the inequalities, we get

$$0 \leq c_3 g(n) \leq f(n) \leq c_4 g(n) \text{ for all } n > n_3.$$

Which is the definition of Θ.

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 $\Omega(g(n))$.

If T_w is the worst-case running time and T_b is the best-case running time, we know that

$$\begin{aligned} 0 \le c_1 g(n) \le T_b(n) & \qquad \text{for } n > n_b \\ \text{and } 0 \le T_w(n) \le c_2 g(n) & \qquad \text{for } n > n_w. \end{aligned}$$

Combining them we get

$$0 \le c_1 g(n) \le T_b(n) \le T_w(n) \le c_2 g(n) \text{ for } n > \max(n_b, n_w).$$

Since the running time is bound between T_b and T_w and the above is the definition of the Θ-notation, proved.

3.1-7

> Prove $o(g(n)) \cap w(g(n))$ is the empty set.

Let $f(n) = o(g(n)) \cap w(g(n))$. We know that for any $c_1 > 0$, $c_2 > 0$,

$$\begin{aligned} \exists n_1 > 0 : 0 \le f(n) < c_1 g(n) \\ \text{and } \exists n_2 > 0 : 0 \le c_2 g(n) < f(n). \end{aligned}$$

If we pick $n_0 = \max(n_1, n_2)$, and let $c_1 = c_2$, from the problem definition we get

$$c_1 g(n) < f(n) < c_1 g(n).$$

There is no solutions, which means that the intersection is the empty set.

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 function $g(n, m)$ we denote $O(g(n, m))$ the set of functions:
>
> $$\begin{aligned} O(g(n, m)) = \{f(n, m) : & \text{ there exist positive constants } c, n_0, \text{ and } m_0 \\ & \text{ such that } 0 \le f(n, m) \le cg(n, m) \\ & \text{ for all } n \ge n_0 \text{ or } m \ge m_0. \} \end{aligned}$$
>
> Give corresponding definitions for $\Omega(g(n, m))$ and $\Theta(g(n, m))$.

$$\begin{aligned} \Omega(g(n, m)) = \{f(n, m) : & \text{ there exist positive constants } c, n_0, \text{ and } m_0 \text{ such that} \\ & 0 \le cg(n, m) \le f(n, m) \text{ for all } n \ge n_0 \text{ and } m \ge m_0. \} \end{aligned}$$

$$\begin{aligned} \Theta(g(n, m)) = \{f(n, m) : & \text{ there exist positive constants } c_1, c_2, n_0, \text{ and } m_0 \text{ such that} \\ & 0 \le c_1 g(n, m) \le f(n, m) \le c_2 g(n, m) \text{ for all } n \ge n_0 \text{ and } m \ge m_0. \} \end{aligned}$$

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.

$$\begin{aligned} f(m) \le f(n) &\quad \text{for } m \le n \\ g(m) \le g(n) &\quad \text{for } m \le n, \\ \to f(m) + g(m) \le f(n) + g(n), \end{aligned}$$

which proves the first function.

Then

$$f(g(m)) \le f(g(n)) \text{ for } m \le n.$$

This is true, since $g(m) \le g(n)$ and $f(n)$ is monotonically increasing.

If both functions are nonnegative, then we can multiply the two equalities and we get

$$f(m) \cdot g(m) \le f(n) \cdot g(n).$$

3.2-2

> Prove equation (3.16).

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

3.2-3

> Prove equation (3.19). Also prove that $n! \ne \omega(2^n)$ and $n! \ne o(n^n)$.
>
> $$\lg(n!) = \Theta(n \lg n) \tag{3.19}$$

We can use **Stirling's approximation** to prove these three equations.

For equation (3.19),

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

For $n! \ne \omega(2^n)$,

$$\begin{aligned}
\lim_{n\to\infty}\frac{2^n}{n!} &= \lim_{n\to\infty}\frac{2^n}{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1+\Theta\left(\frac{1}{n}\right)\right)} \\
&= \lim_{n\to\infty}\frac{1}{\sqrt{2\pi n}\left(1+\Theta\left(\frac{1}{n}\right)\right)}\left(\frac{2e}{n}\right)^n \\
&\le \lim_{n\to\infty}\left(\frac{2e}{n}\right)^n \\
&\le \lim_{n\to\infty}\frac{1}{2^n} = 0,
\end{aligned}$$

where the last step holds for $n > 4e$.

For $n! \ne o(n^n)$,

$$\begin{aligned}
\lim_{n\to\infty}\frac{n^n}{n!} &= \lim_{n\to\infty}\frac{n^n}{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1+\Theta\left(\frac{1}{n}\right)\right)} \\
&= \lim_{n\to\infty}\frac{e^n}{\sqrt{2\pi n}\left(1+\Theta\left(\frac{1}{n}\right)\right)} \\
&= \lim_{n\to\infty} O(\frac{1}{\sqrt{n}})e^n \\
&\ge \lim_{n\to\infty}\frac{e^n}{c\sqrt{n}} && \text{(for some constant } c > 0) \\
&\ge \lim_{n\to\infty}\frac{e^n}{cn} \\
&= \lim_{n\to\infty}\frac{e^n}{c} = \infty.
\end{aligned}$$

3.2-4 ⋆

> Is the function $\lceil \lg n \rceil!$ polynomially bounded? Is the function $\lceil \lg \lg n \rceil!$ polynomially bounded?

Proving that a function $f(n)$ is polynomially bounded is equivalent to proving that $\lg(f(n)) = O(\lg n)$ for the following reasons.

- If f is polynomially bounded, then there exist constants c, k, n_0 such that for all $n \ge n_0$, $f(n) \le cn^k$. Hence, $\lg(f(n)) \le kc \lg n$, which means that $\lg(f(n)) = O(\lg n)$.
- If $\lg(f(n)) = O(\lg n)$, then f is polynomially bounded.

In the following proofs, we will make use of the following two facts:

1. $\lg(n!) = \Theta(n \lg n)$
2. $\lceil \lg n \rceil = \Theta(\lg n)$

$\lceil \lg n \rceil!$ is not polynomially bounded because

$$\begin{aligned}
\lg(\lceil \lg n \rceil !) &= \Theta(\lceil \lg n \rceil \lg \lceil \lg n \rceil) \\
&= \Theta(\lg n \lg \lg n) \\
&= \omega(\lg n) \\
&\neq O(\lg n).
\end{aligned}$$

$\lceil \lg \lg n \rceil !$ is polynomially bounded because

$$\begin{aligned}
\lg(\lceil \lg \lg n \rceil !) &= \Theta(\lceil \lg \lg n \rceil \lg \lceil \lg \lg n \rceil) \\
&= \Theta(\lg \lg n \lg \lg \lg n) \\
&= o((\lg \lg n)^2) \\
&= o(\lg^2(\lg n)) \\
&= o(\lg n) \\
&= O(\lg n).
\end{aligned}$$

The last step above follows from the property that any polylogarithmic function grows more slowly than any positive polynomial function, i.e., that for constants $a, b > 0$, we have $\lg^b n = o(n^a)$. Substitute $\lg n$ for n, 2 for b, and 1 for a, giving $\lg^2(\lg n) = o(\lg n)$.

Therefore, $\lg(\lceil \lg \lg n \rceil !) = O(\lg n)$, and so $\lceil \lg \lg n \rceil !$ is polynomially bounded.

3.2-5 ⋆

> Which is asymptotically larger: $\lg(\lg^* n)$ or $\lg^*(\lg n)$?

We have $\lg^* 2^n = 1 + \lg^* n$,

$$\begin{aligned}
\lim_{n\to\infty} \frac{\lg(\lg^* n)}{\lg^*(\lg n)} &= \lim_{n\to\infty} \frac{\lg(\lg^* 2^n)}{\lg^*(\lg 2^n)} \\
&= \lim_{n\to\infty} \frac{\lg(1 + \lg^* n)}{\lg^* n} \\
&= \lim_{n\to\infty} \frac{\lg(1 + n)}{n} \\
&= \lim_{n\to\infty} \frac{1}{1 + n} \\
&= 0.
\end{aligned}$$

Therefore, we have that $\lg^*(\lg n)$ is asymptotically larger.

3.2-6

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

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

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.

- Base case

  For $i = 0$,

  $$
  \begin{aligned}
  \frac{\phi^0 - \hat{\phi}^0}{\sqrt{5}} &= \frac{1-1}{\sqrt{5}} \\
  &= 0 \\
  &= F_0.
  \end{aligned}
  $$

  For $i = 1$,

  $$
  \begin{aligned}
  \frac{\phi^1 - \hat{\phi}^1}{\sqrt{5}} &= \frac{(1+\sqrt{5}) - (1-\sqrt{5})}{2\sqrt{5}} \\
  &= 1 \\
  &= F_1.
  \end{aligned}
  $$

- Assume

  - $F_{i-1} = (\phi^{i-1} - \hat{\phi}^{i-1})/\sqrt{5}$ and
  - $F_{i-2} = (\phi^{i-2} - \hat{\phi}^{i-2})/\sqrt{5}$,

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

3.2-8

> Show that $k \ln k = \Theta(n)$ implies $k = \Theta(n / \lg n)$.

From the symmetry of Θ,

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

Let's find $\ln n$,

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

Let's divide the two,

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

Problem 3-1 Asymptotic behavior of polynomials

> Let
>
> $$p(n) = \sum_{i=0}^{d} a_i n^i,$$
>
> where $a_d > 0$, be a degree-d polynomial in n, and let k be a constant. Use the definitions of the asymptotic notations to prove the following properties.
>
> **a.** If $k \ge d$, then $p(n) = O(n^k)$.
>
> **b.** If $k \le d$, then $p(n) = \Omega(n^k)$.
>
> **c.** If $k = d$, then $p(n) = \Theta(n^k)$.
>
> **d.** If $k > d$, then $p(n) = o(n^k)$.
>
> **e.** If $k < d$, then $p(n) = \omega(n^k)$.

Let's see that $p(n) = O(n^d)$. We need do pick $c = a_d + b$, such that

$$\sum_{i=0}^{d} a_i n^i = a_d n^d + a_{d-1} n^{d-1} + \cdots + a_1 n + a_0 \leq cn^d.$$

When we divide by n^d, we get

$$c = a_d + b \geq a_d + \frac{a_{d-1}}{n} + \frac{a_{d-2}}{n^2} + \cdots + \frac{a_0}{n^d}.$$

and

$$b \geq \frac{a_{d-1}}{n} + \frac{a_{d-2}}{n^2} + \cdots + \frac{a_0}{n^d}.$$

If we choose $b = 1$, then we can choose n_0,

$$n_0 = \max(da_{d-1}, d\sqrt{a_{d-2}}, \ldots, d\sqrt[d]{a_0}).$$

Now we have n_0 and c, such that

$$p(n) \leq cn^d \quad \text{for } n \geq n_0,$$

which is the definition of $O(n^d)$.

By chosing $b = -1$ we can prove the $\Omega(n^d)$ inequality and thus the $\Theta(n^d)$ inequality.

It is very similar to prove the other inequalities.

Problem 3-2 Relative asymptotic growths

> Indicate for each pair of expressions (A, B) in the table below, whether A is O, o, Ω, ω, or Θ of B. Assume that $k \geq 1$, $\epsilon > 0$, and $c > 1$ are constants. Your answer should be in the form of the table with "yes" or "no" written in each box.

A	B	O	o	Ω	ω	Θ
$\lg^k n$	n^ϵ	yes	yes	no	no	no
n^k	c^n	yes	yes	no	no	no
$\sqrt{n}$	$n^{\sin n}$	no	no	no	no	no
2^n	$2^{n/2}$	no	no	yes	yes	no
$n^{\lg c}$	$c^{\lg n}$	yes	no	yes	no	yes
$\lg(n!)$	$\lg(n^n)$	yes	no	yes	no	yes

Problem 3-3 Ordering by asymptotic growth rates

> **a.** Rank the following functions by order of growth; that is, find an arrangement $g_1, g_2, \ldots, g_{30}$ of the functions $g_1 = \Omega(g_2), g_2 = \Omega(g_3), \ldots, g_{29} = \Omega(g_{30})$. Partition your list into equivalence classes such that functions $f(n)$ and $g(n)$ are in the same class if and only if $f(n) = \Theta(g(n))$.
>
> | | | | | | |
> |---|---|---|---|---|---|
> | $\lg(\lg^* n)$ | $2^{\lg^* n}$ | $(\sqrt{2})^{\lg n}$ | n^2 | $n!$ | $(\lg n)!$ |
> | $(\frac{3}{2})^n$ | n^3 | $\lg^2 n$ | $\lg(n!)$ | 2^{2^n} | $n^{1/\lg n}$ |
> | $\lg\lg n$ | $\lg^* n$ | $n \cdot 2^n$ | $n^{\lg\lg n}$ | $\lg n$ | 1 |
> | $2^{\lg n}$ | $(\lg n)^{\lg n}$ | e^n | $4^{\lg n}$ | $(n+1)!$ | $\sqrt{\lg n}$ |
> | $\lg^*(\lg n)$ | $2^{\sqrt{2\lg n}}$ | n | 2^n | $n\lg n$ | $2^{2^{n+1}}$ |
>
> **b.** Give an example of a single nonnegative function $f(n)$ such that for all functions $g_i(n)$ in part (a), $f(n)$ is neither $O(g_i(n))$ nor $\Omega(g_i(n))$.

$2^{2^{n+1}}$
2^{2^n}
$(n+1)!$
$n!$
e^n
$n \cdot 2^n$
2^n
$(3/2)^n$
$(\lg n)^{\lg n} = n^{\lg\lg n}$
$(\lg n)!$
n^3
$n^2 = 4^{\lg n}$
$n\lg n$ and $\lg(n!)$
$n = 2^{\lg n}$
$(\sqrt{2})^{\lg n} (= \sqrt{n})$
$2^{\sqrt{2\lg n}}$
$\lg^2 n$
$\ln n$
$\sqrt{\lg n}$
$\ln\ln n$
$2^{\lg^* n}$
$\lg^* n$ and $\lg^*(\lg n)$
$\lg(\lg^* n)$
$n^{1/\lg n} (= 2)$ and 1

b. For example,

$$f(n) = \begin{cases} 2^{2^{n+2}} & \text{if } n \text{ is even,} \\ 0 & \text{if } n \text{ is odd.} \end{cases}$$

for all functions $g_i(n)$ in part (a), $f(n)$ is neither $O(g_i(n))$ nor $\Omega(g_i(n))$.

Problem 3-4 Asymptotic notation properties

> Let $f(n)$ and $g(n)$ by asymptotically positive functions. Prove or disprove each of the following conjectures.
>
> **a.** $f(n) = O(g(n))$ implies $g(n) = O(f(n))$.
>
> **b.** $f(n) + g(n) = \Theta(\min(f(n), g(n)))$.
>
> **c.** $f(n) = O(g(n))$ implies $\lg(f(n)) = O(\lg(g(n)))$, where $\lg(g(n)) \ge 1$ and $f(n) \ge 1$ for all sufficiently large n.
>
> **d.** $f(n) = O(g(n))$ implies $2^{f(n)} = O(2^{g(n)})$.
>
> **e.** $f(n) = O((f(n))^2)$.
>
> **f.** $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$.
>
> **g.** $f(n) = \Theta(f(n/2))$.
>
> **h.** $f(n) + o(f(n)) = \Theta(f(n))$.

a. Disprove, $n = O(n^2)$, but $n^2 \ne O(n)$.

b. Disprove, $n^2 + n \ne \Theta(\min(n^2, n)) = \Theta(n)$.

c. Prove, because $f(n) \ge 1$ after a certain $n \ge n_0$.

$$\begin{aligned} \exists c, n_0 : \forall n \ge n_0, 0 \le f(n) \le cg(n) \\ \Rightarrow 0 \le \lg f(n) \le \lg(cg(n)) = \lg c + \lg g(n). \end{aligned}$$

We need to prove that

$$\lg f(n) \le d \lg g(n).$$

We can find d,

$$d = \frac{\lg c + \lg g(n)}{\lg g(n)} = \frac{\lg c}{\lg g(n)} + 1 \le \lg c + 1,$$

where the last step is valid, because $\lg g(n) \ge 1$.

d. Disprove, because $2n = O(n)$, but $2^{2n} = 4^n \ne O(2^n)$.

e. Prove, $0 \le f(n) \le cf^2(n)$ is trivial when $f(n) \ge 1$, but if $f(n) < 1$ for all n, it's not correct. However, we don't care this case.

f. Prove, from the first, we know that $0 \le f(n) \le cg(n)$ and we need to prove that $0 \le df(n) \le g(n)$, which is straightforward with $d = 1/c$.

g. Disprove, let's pick $f(n) = 2^n$. We will need to prove that

$$\exists c_1, c_2, n_0 : \forall n \ge n_0, 0 \le c_1 \cdot 2^{n/2} \le 2^n \le c_2 \cdot 2^{n/2},$$

which is obviously untrue.

h. Prove, let $g(n) = o(f(n))$. Then

$$\exists c, n_0 : \forall n \ge n_0, 0 \le g(n) < cf(n).$$

We need to prove that

$$\exists c_1, c_2, n_0 : \forall n \ge n_0, 0 \le c_1 f(n) \le f(n) + g(n) \le c_2 f(n).$$

Thus, if we pick $c_1 = 1$ and $c_2 = c + 1$, it holds.

Problem 3-5 Variations on O and Ω

> Some authors define Ω in a slightly different way than we do; let's use $\overset{\infty}{\Omega}$ (read "omega infinity") for this alternative definition. We say that $f(n) = \overset{\infty}{\Omega}(g(n))$ if there exists a positive constant c such that $f(n) \ge cg(n) \ge 0$ for infinitely many integers n.
>
> **a.** Show that for any two functions $f(n)$ and $g(n)$ that are asymptotically nonnegative, either $f(n) = O(g(n))$ or $f(n) = \overset{\infty}{\Omega}(g(n))$ or both, whereas this is not true if we use Ω in place of $\overset{\infty}{\Omega}$.
>
> **b.** Describe the potential advantages and disadvantages of using $\overset{\infty}{\Omega}$ instead of Ω to characterize the running times of programs.
>
> Some authors also define O in a slightly different manner; let's use O' for the alternative definition. We say that $f(n) = O'(g(n))$ if and only if $|f(n)| = O(g(n))$.
>
> **c.** What happens to each direction of the "if and only if" in Theorem 3.1 if we substitute O' for O but we still use Ω?
>
> Some authors define $\tilde{O}$ (read "soft-oh") to mean O with logarithmic factors ignored:
>
> $$\begin{aligned} \tilde{O}(g(n)) = \{f(n) : & \text{ there exist positive constants } c, k, \text{ and } n_0 \text{ such that} \\ & 0 \le f(n) \le cg(n) \lg^k(n) \text{ for all } n \ge n_0 .\} \end{aligned}$$
>
> **d.** Define $\tilde{\Omega}$ and $\tilde{\Theta}$ in a similar manner. Prove the corresponding analog to Theorem 3.1.

a. We have

$$f(n) = \begin{cases} O(g(n)) \text{ and } \Omega^{\infty}(g(n)) & \text{if } f(n) = \Theta(g(n)), \\ O(g(n)) & \text{if } 0 \le f(n) \le cg(n), \\ \Omega^{\infty}(g(n)) & \text{if } 0 \le cg(n) \le f(n), \text{ for infinitely many integers n.} \end{cases}$$

If there are only finite n such that $f(n) \ge cg(n) \ge 0$. When $n \to \infty$, $0 \le f(n) \le cg(n)$, i.e., $f(n) = O(g(n))$.

Obviously, it's not hold when we use Ω in place of Ω^{∞} .

b.

- Advantages: We can characterize all the relationships between all functions.
- Disadvantages: We cannot characterize precisely.

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

But the conversion is not true.

d. We have

$$\begin{aligned}\tilde{\Omega}(g(n)) = \{f(n) : & \text{ there exist positive constants c, k, and } n_0 \text{ such that} \\ & 0 \le cg(n)\lg^k(n) \le f(n) \text{ for all } n \ge n_0.\}\end{aligned}$$

$$\begin{aligned}\tilde{\Theta}(g(n)) = \{f(n) : & \text{ there exist positive constants } c_1, c_2, k_1, k_2, \text{ and } n_0 \text{ such that} \\ & 0 \le c_1 g(n)\lg^{k_1}(n) \le f(n) \le c_2 g(n)\lg^{k_2}(n) \text{ for all } n \ge n_0.\}\end{aligned}$$

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

Problem 3-6 Iterated functions

> We can apply the iteration operator * used in the $\lg^*$ function to any monotonically increasing function $f(n)$ over the reals. For a given constant $c \in \mathbb{R}$, we define the iterated function $f_c{}^*$ by $f_c{}^*(n) = \min\{i \ge 0 : f^{(i)}(n) \le c\}$ which need not be well defined in all cases. In other words, the quantity $f_c{}^*(n)$ is the number of iterated applications of the function f required to reduce its argument down to c or less.

For each of the following functions $f(n)$ and constants c, give as tight a bound as possible on $f_c{}^*(n)$.

$f(n)$	c	f_c^*
$n - 1$	0	$\Theta(n)$
$\lg n$	1	$\Theta(\lg^* n)$
$n/2$	1	$\Theta(\lg n)$
$n/2$	2	$\Theta(\lg n)$
$\sqrt{n}$	2	$\Theta(\lg \lg n)$
$\sqrt{n}$	1	does not converge
$n^{1/3}$	2	$\Theta(\log_3 \lg n)$
$n/ \lg n$	2	$\omega(\lg \lg n), o(\lg n)$