3.1-1

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

Step 1 of 2

Let's max function defined as the function h(n) = max(f (n), g(n)).

Then $$h(n) = \begin{cases} f(n) & \text{if } f(n) \geq g(n) \\ g(n) & \text{if } f(n) \prec g(n) \end{cases}$$

Since f (n) and g(n) are asymptotically nonnegative, there exists n_0 such that f (n) ≥ 0 and g(n) ≥ 0 for all n ≥ n0. Thus for n ≥ n0, f (n) + g(n) ≥ f (n) ≥ 0 and f (n)+g(n) ≥ g(n) ≥ 0.

Since for any particular n, h(n) is either f (n) or g(n), we have f (n) + g(n) ≥ h(n) ≥ 0, which shows that h(n) = max(f (n), g(n)) ≤ c2(f (n) + g(n)) for all n ≥ n0 (with c2 = 1 in the definition of Θ).

Step 2 of 2

Similarly, since for any particular n, h(n) is the larger of f (n) and g(n), we have for all n ≥ n0, 0 ≤ f (n) ≤ h(n) and 0 ≤ g(n) ≤ h(n). Adding these two inequalities yields 0 ≤ f (n) + g(n) ≤ 2h(n), or equivalently 0 ≤ (f (n) + g(n))/2 ≤ h(n), which shows that h(n) = max(f(n), g(n)) ≥ c1(f (n)+g(n)) for all n ≥ n0 (with c1 = 1/2 in the definition of Θ).

3.1-2

Show that for any real constants a and b, where $b > 0$,

$$(n + a)^b = \Theta(n^b) . \quad (3.2)$$

Step 1 of 2

For any real constants a and b, b>0

$$(n+a)^b = \theta(n^b)$$

Step 2 of 2

From the basic definition

$$0 \leq C_1 n^b \leq (n+a)^b \leq C_2 n^b \text{ for all } n \geq n_0$$

Selecting C_2 should be greater than

$$C_2 \geq \frac{(n+a)^b}{n^b}$$

$$C_2 \geq \left(\frac{n+a}{n}\right)^b$$

$$C_1 \leq \left(\frac{n+a}{n}\right)^b$$

We need to selected C_1 such that it should be $\left(\frac{n+a}{n}\right)^b - 1$ where as C_2 must be $\left(\frac{n+a}{n}\right)^b + 1$ for any real constants a , b and $b > 0$ thus the condition holds

3.1-3

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

Step 1 of 1

The given statement is "The running time of algorithm A is at least O (n^2)" is meaning less" because let us assume that the running of algorithm be T (n).This T (n) $\geq O(n^2)$ means that T (n) $\geq f(n)$ for some function in $O(n^2)$. This is true for any algorithm, since the function g(n) =0 for any n is in $O(n^2)$ and run times are always >0.So,the given statement tells that nothing about the running time.

3.1-4

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

Step 1 of 1

the first statement is true, and the second is false.

Observe that

$$\begin{aligned} 2^{n+1} &= 2 \cdot 2^n \\ &= \text{const} \cdot 2^n \\ &= O(2^n) \end{aligned}$$

Also note

$$\begin{aligned} 2^{2n} &= (2^2)^n \\ &= 4^n \\ &= \omega(2^n) \end{aligned}$$

hence $2^{2n} \neq O(2^n)$ since $\omega(2^n) \cap O(2^n) = \varnothing$.

3.1-5

Prove Theorem 3.1.

Step 1 of 2

Theorem: for any two functions f(n) and g(n), we have $f(n) = \theta(g(n))$ if and only if $f(n) = 0(g(n))$ and $f(n) = \Omega(g(n))$

Proof: Simply to say that, as the $\theta(g(n))$

Definition implies that

$0 \leq C_1(g(n)) \leq f(n) \leq C_2(g(n))$

Means it should satisfy both upper and lower bounds. For example $an^2 + bn + c = \theta(n^2)$ for any constants a,b, and , where a>0. immediately implies that $an^2 + bn + c = \Omega(n^2)$ and $an^2 + bn + c = 0(n^2)$

Since Ω - notation describes a lower bound, when we use it to bound the best-case running time of an algorithm. So the running time of an algorithm must fall between the $\Omega(n)$ and $0(n^2)$. Which denotes lower bound and upper bounds respectively. Which are nothing but the bounds specified by the 'Θ'

Step 2 of 2

Notation:

i.e $\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$

$$f(n) = \Theta(g(n)) \Rightarrow f(n) = O(g(n))$$

$$\Theta(g(n)) < O(g(n)) \qquad \ldots\ldots (1)$$

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

$$\Theta(g(n)) \prec \Omega(g(n)) \qquad \ldots\ldots (2)$$

From (1) & (2) we can say $f(n) = \theta(g(n))$

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 $\Omega(g(n))$.

Step 1 of 2

From the definition of $\Theta(g(n))$, consider the inequality

$$0 \le C_1(g(n)) \le f(n) \le C_2(g(n))$$

This inequality should satisfy both upper bound and lower bound.

This inequality can be split into two inequalities: $0 \le f(n) \le cg(n)$ and $0 \le cg(n) \le f(n)$.

Now it is enough if proved that for an algorithm the worst-case running time is $O(g(n))$ and the best-case running time is $\Omega(g(n))$.

For example $an^2 + bn + c = \Theta(n^2)$ where *a, b, c* are constants and $a > 0$. This directly implies that $an^2 + bn + c = \Omega(n^2)$ and $an^2 + bn + c = O(n^2)$

Ω can be used to represent the best-case running time, as it denote the asymptotic lower bound. So $0 \le cg(n) \le f(n)$ represents $\Omega(g(n))$.

O can be used to represent the worst-case running time, as it denote the asymptotic upper bound. So $0 \le f(n) \le cg(n)$ represents $O(g(n))$.

So the running time of an algorithm should be between $\Omega(n^2)$ and $O(n^2)$, and they denote lower bound and upper bounds respectively.

Step 2 of 2

Now considering the worst-case running time as $O(g(n))$ and the best-case running time as $\Omega(g(n))$, it is enough to prove that the running time of an algorithm is $\Theta(g(n))$.

$$\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$$

$$f(n) = \Theta(g(n)) \Rightarrow f(n) = O(g(n))$$

$$\Theta(g(n)) < O(g(n)) \text{ ...... (1)}$$

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

$$\Theta(g(n)) < \Omega(g(n)) \text{ ...... (2)}$$

From (1) & (2) it can be said $f(n) = \Theta(g(n))$

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

3.1-7

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

Step 1 of 3

$O(g(n)) \cap w(g(n))$ is the empty set

Proof: In order to prove this $O(g(n))$ denotes the upper bound that is not asymptotically tight

$O(g(n)) = \{f(n)$: for any positive constant C>0, there exist a constant $n_0 > 0$ such that $0 < f(n) \le cg(n)$, for all $n \ge n_0\}$

Provide feedback (0)

Step 2 of 3

$\Omega(g(n))$ denotes the lower bound that is not asymptotically tight $0 \le cg(n) < f(n)$

$f(n) \in \Omega(g(n))$ iff $g(n) \in O(f(n))$

$\Omega(g(n))$ $\{f(n)$: for any positive constant c>0, there exists a constant $n_0 > 0$ such that $0 < f(n) \le cg(n) < f(n)$ which leads to a f(n) < f(n) concludes for all $n \ge n_0$ thus no f(n) exists, and $O(g(n)) \cap w(g(n))$ is empty set}

Step 3 of 3

Consider two real numbers a,b Then

$f(n) = O(g(n)) \approx a < b$

$f(n) = \Omega(g(n)) \approx a > b$

If we take a=2 and b=3 the set that contains elements satisfying a<b will not be satisfied by a>b thus the intersection of

$O(g(n)) \cap \Omega(g(n)) = \phi$ (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 by $O(g(n,m))$ the set of functions

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

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

Step 1 of 3

$O(g(n))=\{f(n,m)$: there exist positive constants c, n_0 and m_0 such that $0 \le f(n,m) \le C_g(n,m)$ for all $n \ge n_0$ and $m \ge m_0\}$

Provide feedback (0)

Step 2 of 3

The corresponding definition for

$\Omega(g(n,m))=\{$f(n,m): there exists positive constants C , n_0 and m_0 such that $0 \le Cg(n,m) \le f(n,m)$ whenever $n \ge n_0$ and $m > m_0\}$

Provide feedback (0)

Step 3 of 3

The corresponding definition for

$\theta(g(n,m))=\{$f(n,m): there exists positive constants C_1, C_2 and m_0, n_0 such that $0 \le C_1 g(n,m) \le f(n,m) \le C_2 g(n,m)$ for all $n \ge n_0$ and $m \ge m_0\}$

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.

Step 1 of 3

We have to prove that if $f(n)$ and $g(n)$ are monotonically increasing functions then their sum $f(n)+g(n)$ is also monotonically increasing function.

Since $f(n)$ and $g(n)$ are monotonically increasing functions means if $n<m$ $f(n)\le f(m)$ and $g(n)\le g(m)$ ......(1)

We will prove this by contradiction.
Let us assume that $n<m$ and $f(n)+g(n)\ge f(m)+g(m)$
From (1):

$$f(n)\le f(m)$$
$$f(n)+g(n)\le f(m)+g(n) \qquad \text{(adding } g(n) \text{ both sides)}$$
$$f(n)+g(n)\le f(m)+g(m) \qquad \text{(Since } g(n)\le g(m)\text{)}$$

This is a contradiction to our assumption $n<m$ and $f(n)+g(n)\ge f(m)+g(m)$.
Therefore, if $f(n)$ and $g(n)$ are monotonically increasing functions then their sum $f(n)+g(n)$ is also monotonically increasing function.

Step 2 of 3

Now we have to prove that if $f(n)$ and $g(n)$ are monotonically increasing functions then $f(g(n))$ is also monotonically increasing function.

Since $f(n)$ and $g(n)$ are monotonically increasing functions means if $n<m$ $f(n)\le f(m)$ and $g(n)\le g(m)$ ......(1)

We will prove this by contradiction.
Let us assume that $n<m$ and $f(g(n))>f(g(m))$
Let $x=g(n)$, $y=g(m)$.

$$\Rightarrow x\le y \qquad [\text{From (1)}]$$

Clearly, $f(x)\le f(y)\Rightarrow f(g(n))\le f(g(m))$ this is a contradiction to our assumption.

Therefore, if $f(n)$ and $g(n)$ are monotonically increasing functions then $f(g(n))$ is also monotonically increasing function.

Step 3 of 3

Now we have to prove that if $f(n)$ and $g(n)$ are monotonically increasing and are in addition nonnegative functions then $f(n) \cdot g(n)$ is also monotonically increasing function.

Since $f(n)$ and $g(n)$ are monotonically increasing functions means if $n < m$ $f(n) \le f(m)$ and $g(n) \le g(m)$ ......(1)

We will prove this by contradiction.
Let us assume that $n < m$ and $f(n) \cdot g(n) > f(m) \cdot g(m)$.
Now, $f(n) \le f(m)$ and $g(n) \le g(m)$

From (1):

$$f(n) \le f(m)$$

$$f(n) \cdot g(n) \le f(m) \cdot g(n) \qquad \text{(multiply } g(n) \text{ both sides)} \qquad \text{......(2)}$$

$$f(n) \cdot g(n) \le f(m) \cdot g(m) \qquad \text{(Since } g(n) \le g(m)\text{)} \qquad \text{......(3)}$$

Here (2) and (3) holds since $f(n)$ and $g(n)$ are monotonically increasing and are in addition nonnegative functions.
This is a contradiction to our assumption $n < m$ and $f(n) \cdot g(n) > f(m) \cdot g(m)$.

Therefore, if $f(n)$ and $g(n)$ are monotonically increasing and are in addition non-negative functions, then $f(n) \cdot g(n)$ is also monotonically increasing function.

3.2-2

Prove equation (3.16).

Step 1 of 1

Prove $a^{\log_b^c} = c^{\log_b^a}$

Let $a^{\log_b^c} = t$ --- (1)

Applying log on both sides

$$\log a^{\log_b^c} = \log t$$

$$\log_b^c \log a = \log t \qquad \left| \because \log a^m = m \log a \right|$$

$$\frac{\log c}{\log b} \log a = \log t \qquad \left| \because \log_b^a = \frac{\log a}{\log b} \right|$$

$$\frac{\log a}{\log b} \log c = \log t$$

$$\log_b^a \log c = \log t \qquad \left| \because \frac{\log a}{\log b} = \log_b^a \right|$$

$$\log c^{\log_b^a} = \log t \qquad \left| \because m \log a = \log_a^m \right|$$

$$c^{\log_b^a} = t \qquad \text{--- (2)}$$

From (1) & (2) $\boxed{a^{\log_b^c} = c^{\log_b^a}}$

3.2-3

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

Step 1 of 3

Consider the equation:

$$\lg(n!) = \Theta(n \lg n)$$

The mathematical proof:

$$\lg n! = \lg 1 + \lg 2 + \dots\dots\dots + \lg n$$

$$= \sum_{k=1}^{n} \lg k$$

$$\approx \int_1^{\infty} \lg x dx$$

$$= [x \lg x - x]_1^n \qquad \left[Since \ \left[F^1(x)\right]_a^b = \left(F^1(b) - F^1(a)\right) \right]$$

$$= [(n \lg n - n) - (1.\log 1 - 1)] \qquad [Since \ \ \log 1 = 0]$$

$$= n \lg n - n + 1$$

$$\approx n \lg n - n$$

$$= \Theta(n \lg n)$$

Thus, $\boxed{\lg(n!) = \Theta(n \lg n)}$

Step 2 of 3

The equation can be proved by using Stirling's approximation:

The equation holds for all $n \geq 1$:

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n .e^{\alpha_n}$$

$$= \sqrt{2\pi}.n^{1/2}.\frac{n^n}{e^n}.e^{\alpha_n}$$

$$= \sqrt{2\pi}.n^{1/2}.n^n.\frac{1}{e^n}.e^{\alpha_n} \qquad \left[Since, \ x^a + x^b = x^{a+b}\right]$$

$$= \sqrt{2\pi}.n^{1/2+n}.e^{\alpha_n - n} \qquad \left[Since, \ x^a + x^b = x^{a+b}, \frac{x^a}{x^b} = x^{a-b}\right]$$

$$n! = \sqrt{2\pi} \ n^{n+1/2} \ e^{-n} \qquad \left[Since, \ e^{\alpha_n} = 0 \ \ \text{constant}\right]$$

Apply log both sides then

$$\lg n! = \lg(\sqrt{2\pi}) + \lg n^{n+1/2} + \lg e^{\alpha_n - n}$$

$$= \lg(\sqrt{2\pi}) + (n + 1/2)\log n + (\alpha_n - n)\lg e$$

$$= \lg(\sqrt{2\pi}) + n \lg n + 1/2 \lg n - n \qquad \therefore \log_e^e = 1$$

$$\approx n \lg n - n$$

$$= \Theta(n \lg n)$$

Step 3 of 3

Consider the equation:

$n! = o(n^n)$

Apply limit theorem then we get

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

隐藏空白

Since the limit is equal to zero, $\log_2 n$ has a smaller order of growth than $\sqrt{n}$. So, $\lim_{n\to\infty}\frac{n!}{n^n} = 0$,we can use the little-oh notation.

Thus, $n! \in o(n^n)$

The given equation:

$n! = \omega\left(2^n\right)$

Apply Limit theorem then we get

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

Thus 2^n grows very fast, $n!$ grows still faster. We can write symbolically that $n! \in \omega(2^n)$

3.2-4 ★

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

Step 1 of 3

A function $f(n)$ is **polynomially bounded** if $f(n) = O(n^k)$ for some constant k.

A function $f(n)$ is polynomially bounded is equivalent to proving that

$\lg(f(n)) = O(\lg(n))$ for the following reasons:

1. If $f(n)$ is polynomially bounded, then there exist constants c, k, n_0 such that

for all $n \geq n_0$ · $f(n) \leq cn^k$.

Hence, $\lg(f(n)) \leq k.c\lg(n)$ where c and k are constants

$$\lg(f(n)) = O(\lg(n))$$

2. Similarly, if $\lg(f(n)) = O(\lg(n))$, then f (n) is polynomially bounded.

Step 2 of 3

The essential proofs are

$$\lg(n!) = \Theta(n \lg n)$$

The equation can be proved by using Stirling's approximation:

The equation holds for all $n \geq 1$:

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n .e^{\alpha_n}$$

$$= \sqrt{2\pi}.n^{1/2}.\frac{n^n}{e^n}.e^{\alpha_n}$$

$$= \sqrt{2\pi}.n^{1/2}.n^n.\frac{1}{e^n}.e^{\alpha_n} \quad \therefore x^a + x^b = x^{a+b}$$

$$= \sqrt{2\pi}.n^{1/2+n}.e^{\alpha_n - n} \qquad \therefore x^a + x^b = x^{a+b}, \frac{x^a}{x^b} = x^{a-b}$$

$$n! = \sqrt{2\pi}\ n^{n+1/2}\ e^{-n} \qquad \therefore e^{\alpha_n} = 0 \ \text{ constant}$$

Apply log both sides then

$$\lg n! = \lg(\sqrt{2\pi}) + \lg n^{n+1/2} + \lg e^{\alpha_n - n}$$

$$= \lg(\sqrt{2\pi}) + (n+1/2)\log n + (\alpha_n - n)\lg e$$

$$= \lg(\sqrt{2\pi}) + n\lg n + 1/2\lg n - n \qquad \therefore \log_e^{\,e} = 1$$

$$\approx n\lg n - n$$

$$= \Theta(n\lg n)$$

If a function is polynomially bounded, its log is log bounded. We can also observe that

$$\lceil \lg(n) \rceil \leq O(\lg(n))$$
$$\lceil \lg(n) \rceil \geq \Omega(\lg(n))$$

Since $\lg(n) < \lceil \lg(n) \rceil < \lg(n) + 1 \leq 2\lg(n)\ for\ all\ n \geq 2$

Hence, $\lceil \lg(n) \rceil = \Theta(\lg(n))$

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

$$= \Theta(\lg n \lg \lg n)$$

$$= \omega(\lg n)$$

Therefore, $\lg(\lceil \lg n \rceil!) \neq O(\lg n)$.

So,

The function $\lceil \log \rceil!$ is not polynomially bounded.

Step 3 of 3

The given polynomial function is $\lceil \lg \lg n \rceil!$

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

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, $\lg(\lceil \lg \lg n \rceil !) = O(\lg n)$

So, the function $\lceil \lg \lg n \rceil !$ is polynomially bounded.

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

Step **1** of 2

The iterated logarithm function is defined as

$$\lg^* n = \min\{i \geq 0 : \lg^{(i)} n \leq 1\}$$

The given equation is $\lg(\lg^* n)$ or $\lg^*(\lg n)$. This can be rewritten as

$$\lg(\lg^{(i)} n) \text{ or } \lg^{(i)}(\lg n)$$

Assume that $f(n) = \lg(\lg^{(i)} n)$ and $g(n) = \lg^{(i)}(\lg n)$ and apply Limit's theorem, then we get

$$\begin{aligned}
\lim_{n\to\infty} \frac{f'(x)}{g'(x)} &= \frac{\lg(\lg^{(i)} n)}{\lg^{(i)}(\lg n)} \\
&= \frac{\frac{1}{\lg^{(i)} n}}{(i).\lg(\lg n).\frac{1}{\lg n}.\frac{1}{n}} \\
&= \frac{(\lg n)(n)}{(\lg^{(i)} n).(i).\lg(\lg n).} \\
&= \infty
\end{aligned}$$

Thus, though $\lg^*(\lg n)$ grows very fast, $\lg(\lg^* n)$ grows still faster. We can write symbolically that $\lg(\lg^* n) \in \Omega(\lg^*(\lg n))$.

Note: However that while big-omega notation does not preclude the possibility that $\lg(\lg^* n)$ and $\lg^*(\lg n)$ have the same order of growth, the limit computed here certainly does.

Step **2** of 2

Let us assume that

$$n = 2^{65,536}$$

$$\begin{aligned}
\lg(\lg^* n) &= \lg(\lg^* 2^{65,536}) && [\text{Therefore}, \lg^* 2^{65,536} = 5] \\
&= \lg(5) && [\text{Therefore}, \lg 5 = 2.321] \\
&= 2.321
\end{aligned}$$

$$n = 2^{65,536}$$

$$\begin{aligned}
\lg^*(\lg_2 n) &= \lg^*(\lg_2 2^{65,536}) && |\text{Therefore}, \lg_2 2^{65,536} = 65,536| \\
&= \lg^*(65,536) && |\text{Therefore}, \lg^*(65,536) = 4| \\
&= 4
\end{aligned}$$

When compared to $\lg(\lg^* n)$, $\lg^*(\lg n)$ is larger.

3.2-6

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

Step 1 of 3

Given data:

$$\text{Golden ration } \phi = \frac{1+\sqrt{5}}{2}$$
$$= 1.61803$$

$$\text{Golden ratio conjugate } \hat{\phi} = \frac{1-\sqrt{5}}{2}$$
$$= -0.61803$$

The equation is

$$x^2 = x + 1$$

Step 2 of 3

Substitute the **golden ratio** value in the equation:

$$\begin{aligned} LHS &= x^2 \\ &= (1.61803)^2 \\ &= 2.61803 \\ RHS &= x + 1 \\ &= 1.61803 + 1 \\ &= 2.61803 \end{aligned}$$

Hence, **LHS=RHS**

Step 3 of 3

Substitute **golden ratio** conjugate in the equation is

$$\begin{aligned} LHS &= x^2 \\ &= (-0.61803)^2 \\ &= 0.3819 \end{aligned}$$

$$\begin{aligned} RHS &= x + 1 \\ &= -0.61803 + 1 \\ &= 0.38197 \end{aligned}$$

Hence, **LHS=RHS**

Hence, both the golden ratio $\phi = \frac{1+\sqrt{5}}{2}$ and its conjugate $\hat{\phi} = \frac{1-\sqrt{5}}{2}$ are satisfying the equation $x^2 = x + 1$.

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.

Step 1 of 2

The given ith Fibonacci number satisfies the equality:

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

Where ϕ be the golden ratio and $\hat{\phi}$ be its conjugate:

$$\phi = \frac{1+\sqrt{5}}{2}, \ \hat{\phi} = \frac{1-\sqrt{5}}{2}$$

The Fibonacci numbers are defined by the following recurrence:

$$F_0 = 0$$
$$F_1 = 1$$
$$F_n = F_{n-1} + F_{n-2} \quad \text{For } n \geq 2$$

Assume that $F_k = \dfrac{\phi^k - \hat{\phi}^k}{\sqrt{5}}$

Step 2 of 2

By induction,

$$F_n = \frac{\phi^{n-1} - \hat{\phi}^{n-1}}{\sqrt{5}} + \frac{\phi^{n-2} - \hat{\phi}^{n-2}}{\sqrt{5}}$$

To verify an identity, let us assume that

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

So, we conclude that $F_n = \dfrac{\phi^n - \hat{\phi}^n}{\sqrt{5}}$, the induction hypothesis holds for n. By induction, the equality holds for all $n \leq 0$.

3.2-8

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

Step 1 of 3

Given that $k\ln k = \Theta(n)$ implies $k = \Theta(n/\ln n)$

$k\ln k = \Theta(n)$ means $\exists c_1, c_2 > 0, n_0 > 0 : c_1 n \le k\ln k \le c_2 n \quad \forall n \ge n_0$ ......(1)

Divide (1) by $\ln n$ gives that

$$\exists c_1, c_2 > 0, n_0 > 0 : c_1 \frac{n}{\ln n} \le k\frac{\ln k}{\ln n} \le c_2 \frac{n}{\ln n} \quad \forall n \ge n_0$$

Therefore, need to bound $\frac{\ln k}{\ln n}$.

Taking ln on both sides of equation (1) gives that

$$\ln c_1 + \ln n \le \ln k + \ln(\ln k) \le \ln c_2 + \ln n \qquad \text{......(2)}$$

Step 2 of 3

$$\begin{aligned} & \ln c_1 + \ln n \le \ln k + \ln(\ln k) \\ \text{Since} \quad & \le \ln k + \ln k \quad [\text{for big } k,\ \ln(\ln k) \le \ln k] \\ & \le 2\ln k \end{aligned}$$

Divide both sides with $2\ln n$ we have

$$\frac{1}{2} + \frac{\ln c_1}{2\ln n} \le \frac{\ln k}{\ln n}$$

$$\frac{\ln k}{\ln n} \ge \frac{1}{2} \qquad \left[\text{since } \frac{\ln c_1}{2\ln n} \text{ is positive}\right] \qquad \text{......(3)}$$

Since

$$\ln k \le \ln k + \ln(\ln k) \le \ln c_2 + \ln n \qquad [\text{for big } n]$$

We have

$$\frac{\ln k}{\ln n} \le 1 + \frac{\ln c_2}{\ln n} \le 2 \qquad \text{......(4)}$$

Step 3 of 3

From equation (2) and (3), we have

$$\frac{c_1}{2}\frac{n}{\ln n} \le \frac{1}{2}k\frac{\ln k}{\ln n} \le k$$

And from equations (2) and (4), we have

$$\frac{1}{2}k \le k\frac{\ln k}{\ln n} \le c_2\frac{n}{\ln n}$$

This implies $k \le 2c_2 \frac{n}{\ln n}$

Therefore, $\boxed{k = \Theta(n/\ln n)}$

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 $k \geq d$, then $p(n) = O(n^k)$.

b. If $k \leq d$, 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 $p(n) = \omega(n^k)$.

Step 1 of 7

Given polynomial:

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

Where $a_d > 0$, be a degree $-d$ in n, and let k be a constant.

We can rewrite the polynomial as

$$\begin{aligned} p(n) &= \sum_{i=0}^{d} a_i n^i \\ &= n^d \sum_{i=0}^{d} a_i \frac{1}{n^{d-i}} \\ &= n^d \left(a_d + \sum_{i=0}^{d-1} a_i \frac{1}{n^{d-i}} \right) \\ &= n^d \left(a_d + q(n) \right) \qquad \text{where } q(d) = \sum_{i=0}^{d-1} a_i \frac{1}{n^{d-i}} \end{aligned}$$

Consider $\lim_{n\to\infty} q(n)$, we have

$\lim_{n\to\infty} q(n) = 0$, so there must be an integer n_0 such that $|q(n)| < \frac{a_d}{2}$ for all $n \geq n_0$.

We have

$$n^d \left(a_d - \frac{a_d}{2} \right) \leq n^d \left(a_d + q(n) \right) \leq n^d \left(a_d + \frac{a_d}{2} \right), \qquad \text{when } n > n_0.$$

Let $c_1 = \frac{a_d}{2}$ and $c_2 = \frac{3a_d}{2}$, $0 \leq c_1 n^d \leq p(n) \leq c_2 n^d$ for all $n \geq n_0$ ......(1)

Step **2** of 7

(a) If $k \ge d$, then $p(n) = O(n^k)$

For a given function m $g(n)$, we denote $O(g(n))$ is the set of functions given by

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

From the given statement $f(n) = p(n)$ and $g(n) = n^k$, where k is a constant.

Given $k \ge d$

$$\Rightarrow n^d \le n^k$$

$$\Rightarrow c_2 n^d \le c_2 n^k \qquad \text{......(2)}$$

From (1) and (2) we have

$$0 \le p(n) \le c_2 n^d \le c_2 n^k \text{ for all } n \ge n_0$$

Hence from the definition, $\boxed{p(n) = O(n^k)}$.

Step **3** of 7

(b) If $k \le d$, then $p(n) = \Omega(n^k)$

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

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

From the given statement $f(n) = p(n)$ and $g(n) = n^k$, where k is a constant.

Given $k \le d$

$$\Rightarrow n^k \le n^d$$

$$\Rightarrow c_1 n^k \le c_1 n^d \qquad \text{......(3)}$$

From (1) and (3) we have

$$0 \le c_1 n^k \le c_1 n^d \le p(n) \text{ for all } n \ge n_0$$

Hence from the definition, $\boxed{p(n) = \Omega(n^k)}$.

Step **4** of 7

(c) If $k = d$, then $p(n) = \Theta(n^k)$

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

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

From the given statement $f(n) = p(n)$ and $g(n) = n^k$, where k is a constant.

From (1) we have

$$0 \le c_1 n^d \le p(n) \le c_2 n^d \text{ for all } n \ge n_0$$

Hence from the definition, $\boxed{p(n) = \Theta(n^k)}$.

Step **5** of 7

(d) If $k > d$, then $p(n) = o(n^k)$

For a given function $g(n)$, we denote $o(g(n))$ is the set of functions given by

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

Provide feedback (0)

Step **6** of 7

From the given statement $f(n) = p(n)$ and $g(n) = n^k$, where k is a constant.

Given $k > d$

$$\Rightarrow n^d < n^k$$

$$\Rightarrow c_2 n^d < c_2 n^k \qquad \text{......(4)}$$

From (1) and (4) we have

$$0 \le p(n) \le c_2 n^d < c_2 n^k \text{ for all } n \ge n_0$$

Hence from the definition, $\boxed{p(n) = o(n^k)}$.

Step **7** of 7

(e) If $k < d$, then $p(n) = \omega(n^k)$

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

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

From the given statement $f(n) = p(n)$ and $g(n) = n^k$, where k is a constant.

Given $k < d$

$$\Rightarrow n^k < n^d$$

$$\Rightarrow c_1 n^k < c_1 n^d \qquad \text{......(5)}$$

From (1) and (5) we have

$$0 \le c_1 n^k < c_1 n^d \le p(n) \text{ for all } n \ge n_0$$

Hence from the definition, $\boxed{p(n) = \omega(n^k)}$.

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
a.	$\lg^k n$	n^ϵ					
b.	n^k	c^n					
c.	$\sqrt{n}$	$n^{\sin n}$					
d.	2^n	$2^{n/2}$					
e.	$n^{\lg c}$	$c^{\lg n}$					
f.	$\lg(n!)$	$\lg(n^n)$					

Step **1** of 7

Relative asymptotic growths:

	A	B	O	0	Ω	ω	θ
a)	$\lg^k n$	n^ε	YES	YES	NO	NO	NO
b)	n^k	C^n	YES	YES	NO	NO	NO
c)	$\sqrt{n}$	$n^{\sin n}$	NO	NO	NO	NO	NO
d)	2^n	$2^{n/2}$	NO	NO	YES	YES	NO
e)	$n^{\lg c}$	$C^{\lg n}$	YES	NO	YES	NO	YES
f)	$\lg(n!)$	$\lg(n^n)$	YES	NO	YES	NO	YES

L'Hopital's Rule:

In the case where $f(n)$ and $g(n)$ are differentiable functions and $\lim_{n\to\infty} f(n) = \lim_{n\to\infty} g(n) = \infty$, the limit of $\frac{f(n)}{g(n)}$ can be computed as the limit of their derivatives:

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

Step **2** of 7

Justifications:

a. Applying L'Hopital's Rule k times, we get

$$\lim_{n\to\infty} \frac{(\lg n)^k}{n^\varepsilon}$$

$$= \frac{k!}{\epsilon^k \ln^k 2} \lim_{n\to\infty} \frac{1}{n^\epsilon}$$

$$= 0$$

So, $\lg^k n = O(n^\varepsilon)$

Step **3** of 7

b. Again, by $\lceil k \rceil$ applications of L'Hopital's rule,

$$\lim_{n\to\infty} \frac{n^k}{c^n} = 0\,,$$

Hence, $n^k = O(c^n)$.

Provide feedback (0)

Step **4** of 7

c. $\sin n$ is a periodic function and takes the values in the range [-1, 1].
When $\sin n$=1 we have,

$$\lim_{n\to\infty} \frac{n}{n^{1/2}} = \infty$$

When $\sin n$ = -1 we have,

$$\lim_{n\to\infty} \frac{n^{-1}}{n^{1/2}} = 0$$

Step **5** of 7

d. $\lim_{n\to\infty} \frac{2^n}{2^{n/2}} = \lim_{n\to\infty} 2^{n/2}$

$= \infty$

Therefore, $2^n = \omega(2^{n/2})$.

Provide feedback (0)

Step **6** of 7

e. $n^{\lg c} = c^{\lg n}$

Take log on both sides:

$$\log\left(n^{\lg c}\right) = \log\left(c^{\lg n}\right)$$

$$(\log c).(\log n) = (\log n).(\log c)$$

Step **7** of 7

f. $\lg(n^n) = n\lg(n)$,

$\log! = \sum_{i=1}^{n} \lg i \le \sum_{i=1}^{n} \lg n = n\lg n$ and thus $\lg! = O(n\lg n)$

$$\begin{aligned} \log! &= \sum_{i=1}^{n} \lg i \ge \sum_{i=n/2+1}^{n} \lg n/2 \\ &= n/2 \lg n/2 \\ &= (n/2-1)\lg n/2 \end{aligned}$$

And thus $\lg n! = \Omega(n\lg n)$

Finally $\boxed{\lg n! = o(n\lg n) \text{ and } \lg n! = \omega(n\lg n)}$

The limit $\boxed{\lim_{n\to\infty} \frac{\lg n!}{\lg n^n}}$ lies in the interval [1/2, 1].

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 satisfying $g_1 = \Omega(g_2)$, $g_2 = \Omega(g_3)$, ..., $g_{29} = \Omega(g_{30})$. 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}	$n^{1/\lg n}$
$\ln \ln n$	$\lg^* n$	$n \cdot 2^n$	$n^{\lg \lg n}$	$\ln n$	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 $\Omega(g_i(n))$.

Step 1 of 5

(A) Ranking is based on the following facts
Exponential functions grow faster than polynomial functions, which grow faster than logarithmic functions:
The base of a logarithm does not matter asymptotically, but the base of an exponential and the degree of a polynomial do matter
In addition several identities are helpful

Step 2 of 5

1. $(\lg n)^{\lg n} = n^{\lg \lg n}$
2. $2^{\lg n} = n$
3. $2 = n^{1/\lg n}$
4. $2^{\sqrt{2 \lg n}} = n\sqrt{2/\lg n}$
5. $(\sqrt{2})^{\lg n} = \sqrt{n}$
6. $\lim_{n\to\infty}(1+x/n)^n = e^x$

Stirling's approximation bounds are useful in ranking expression with factorials

$n! = \Theta\left(n^{n+1/2}e^{-n}\right)$

$\lg(n!) = \Theta(n \lg n)$

$(\lg n)! = \Theta\left(\lg n\right)^{\lg n+1/2} e^{-\lg n}$

Step 3 of 5

The ranking (listed from left to right by row)

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

Step 4 of 5

(B) Give an example of a single nonnegative function f(n) such that for all functions gi(n) in part (a), f(n) is neither O(gi(n)) nor (gi(n)).

Provide feedback (0)

Step 5 of 5

Answer: $f(n) = (1+\sin n).2^{2^{n+2}} + 2$.

3-4 Asymptotic notation properties

Let $f(n)$ and $g(n)$ be 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 $\lg(g(n)) \geq 1$ and $f(n) \geq 1$ for all sufficiently large n.

d. $f(n) = O(g(n))$ implies $2^{f(n)} = O\left(2^{g(n)}\right)$.

e. $f(n) = O\left((f(n))^2\right)$.

f. $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$.

g. $f(n) = \Theta(f(n/2))$.

h. $f(n) + o(f(n)) = \Theta(f(n))$.

Step 1 of 10

Let f(n) and g(n) be asymptotically positive functions

Provide feedback (0)

Step 2 of 10

(A) False, $f(n)=O(g(n))$ does not imply $g(n)=O(f(n))$ clearly, $n=O(n^2)$ but $n^2 \neq O(n)$

Step 3 of 10

(B) false, $f(n)+g(n)$ is not $\theta(\min(f(n),g(n)))$ as an example notice that $n+1 \neq \Theta(\min(n,1)=\Theta(1))$

Provide feedback (0)

Step 4 of 10

(C) $f(n)=O(g(n))$ implies $\lg(f(n))=O(\lg(g(n)))$, where $\lg(g(n))\geq 1$ and $f(n)\geq 1$ for all sufficiently large n. **True**

Step 5 of 10

Proof:

Assume $f(n)=O(g(n))$. Then there exists $c>0$ and $n_0>0$ such that $0\leq f(n)\leq cg(n)$ for all $n\geq n_0$. The above hypotheses say we can take n_0 large enough to also guarantee $f(n)\geq 1$ and $\lg(g(n))\geq 1$ for $n\geq n_0$. Taking $\log_2$ of the combined inequality $1\leq f(n)\leq cg(n)$ yields for all $n\geq n_0$:(*)

$$0\leq \lg(f(n))\leq \lg(c)+\lg(g(n))=\left(\frac{\lg(c)}{\lg(g(n))}+1\right)\lg(g(n))\leq(\lg(c)+1)\lg(g(n)).$$

The last inequality is a consequence of $\lg(g(n))\geq 1$, for this implies $1\geq\frac{1}{\lg(g(n))}$, which implies $\lg(c)\geq\frac{\lg(c)}{\lg(g(n))}$. Define the constant $b=\lg(c)+1$. Inequality (*) gives $0\leq\lg(f(n))\leq b\lg(g(n))$ for all $n\geq n_0$, showing that $\lg(f(n))=O(\lg(g(n)))$, as claimed.

Step 6 of 10

(D). $f(n)=O(g(n))$ implies $2^{f(n)}=O(2^{g(n)})$. **False**

Counter-Example:

Let $f(n)=2n$ and $g(n)=n$. Then $2^{g(n)}=2^n$ and $2^{f(n)}=2^{2n}=4^n=\omega(2^n)$, so $2^{f(n)}=\omega(2^{g(n)})$, and therefore $2^{f(n)}\neq O(2^{g(n)})$.

Step 7 of 10

(E). $f(n) = O((f(n))^2)$. **False**

Counter-Example:

Let $f(n) = 1/n$. Then $f(n) = \omega((f(n))^2)$ since $\lim_{n\to\infty} \frac{f(n)}{f(n)^2} = \lim_{n\to\infty} \frac{1}{f(n)} = \lim_{n\to\infty} n = \infty$,

and therefore $f(n) \neq O((f(n))^2)$.

Provide feedback (0)

Step 8 of 10

(F). **True** , $f(n) = 0(g(n))$ implies $g(n) = \Omega(f(n))$ we have $f(n) \leq C.g(n)$ for positive C and thus $1/c.f(n) \leq g(n)$

Step 9 of 10

(G). false, clearly $2^n \not\leq C.2^{n/2} = C\sqrt{2^n}$ for any constant C if n is sufficient large

Provide feedback (0)

Step 10 of 10

(H). $f(n) + o(f(n)) = \Theta(f(n))$. **True**

Proof:

In the above formula, $o(f(n))$ stands for some anonymous function $h(n)$ in the class $o(f(n))$, whence $\lim_{n\to\infty} \frac{h(n)}{f(n)} = 0$. Thus $\lim_{n\to\infty} \frac{f(n)+h(n)}{f(n)} = \lim_{n\to\infty}\left(1+\frac{h(n)}{f(n)}\right) = 1$,

and $f(n) + h(n) = \Theta(f(n))$, as claimed.

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 $f(n) \geq cg(n) \geq 0$ 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 still use Ω?

Some authors define $\widetilde{O}$ (read "soft-oh") to mean O with logarithmic factors ignored:

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

d. Define $\widetilde{\Omega}$ and $\widetilde{\Theta}$ in a similar manner. Prove the corresponding analog to Theorem 3.1.

Step 1 of 5

VARIATIONS ON $(\mathrm{O}\, and\, \Omega)$

a .

If condition is that $f(n) \notin \overset{\infty}{\Omega}(g(n))$ and $g(n)$ such that it is asymptotically non-negative for all the positive constants *c*, Such that: $f(n) \le c \times g(n)$ for large value of *n*.

And this condition is followed by the condition $c \times g(n) \ge 0$ and negation of the definition of $f(n) \in \overset{\infty}{\Omega}(g(n))$. Because definition says 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*. By having the discussion the obtained fact is, that either there is the condition of

$f(n) \in \overset{\infty}{\Omega}(g(n))$ *or* $f(n) \in \mathrm{O}(g(n))$.

Step 2 of 5

b .

First discussion is on the omega and the omega infinity notation and then on the basis of this second discussion will be on its advantage or disadvantage,

Omega notation: It is denoted by symbol Ω. It is an asymptotic lower bound notation. It is defined as,

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

Whereas,

$$\overset{\infty}{\Omega}(g(n)) = \left\{ \begin{array}{l} f(n) : \text{there exist positive constants } c \text{ such that} \\ 0 \le cg(n) \le f(n) \text{ for a larger value of } n \end{array} \right\}$$

Advantage: It has a wider range of values for the notations so a large range of the complexity can be analyzed.

Disadvantage: As the range is not fixed so it is not possible to determine its limitation and its consequences to an indefinite analysis.

Step 3 of 5

c.

Consider the theorem 3.1. When O' is substituted in place of O but in the theorem 3.1 Ω is used, then the" if and only if" direction will have modification as:

$$|f(n)| = O(g(n)) \text{and} |f(n)| = \Omega(g(n))$$

Because from the author definition,

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

If and only if

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

Step 4 of 5

d.

Here defining the soft theta and soft omega:

1. $\tilde{\Omega}$ (Soft omega):

$$\tilde{\Omega}(g(n)) = \left\{ \begin{array}{l} f(n) : \text{ there exist positive } c, k, \text{and } n_0 \text{ such that} \\ \qquad f(n) \ge cg(n) \lg^k(n) \text{ for all } n \ge n_0 \end{array} \right\}$$

2. $\tilde{\Theta}$ (Soft theta):

$$\tilde{\Theta}(g(n)) = \left\{ \begin{array}{l} f(n) : \text{ there exist positive } c, k_1, k_2 \text{and } n_0 \text{ such that} \\ \qquad cg(n) \lg^{k_1}(n) \le f(n) \le cg(n) \lg^{k_2}(n) \text{ for all } n \ge n_0 \end{array} \right\}$$

Step **5** of 5

For any two function *f*(*n*) and *g*(*n*), the function $f(n)=\tilde{\Theta}(g(n))$ if and only if $f(n)=\tilde{O}(g(n))$ and $f(n)=\tilde{\Omega}(g(n))$.

These notation are similar to older one only difference is that here the logarithms factor extra added to them. This logarithm function used in the case of larger inputs. From the definition soft-oh is lower bound in nature and soft-omega is upper bound in nature and soft theta lies in between them.

So it is obvious that if any function is equal to soft-theta then that function is also equal to soft theta and soft-oh. Using the theorem 3.1 the condition is found that an asymptotic upper and lower bound is obtained from the asymptotically tight bound.

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

$$f_c^*(n) = \min\left\{i \geq 0 : f^{(i)}(n) \leq c\right\} ,$$

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	$f_c^*(n)$
a.	$n-1$	0	
b.	$\lg n$	1	
c.	$n/2$	1	
d.	$n/2$	2	
e.	$\sqrt{n}$	2	
f.	$\sqrt{n}$	1	
g.	$n^{1/3}$	2	
h.	$n/\lg n$	2	

Step **1** of 2

Iterated functions

$$f_c{}^*(n) = \min\left\{i \geq 0 : f^{(i)}(n) \leq C\right\}$$

Step 2 of 2

	$f(n)$	C	$f_c^*(n)$
$a)$	$n-1$	0	n
$b)$	$\lg n$	1	$\lg n$
$c)$	$n/2$	1	$\lg n$
$d)$	$n/2$	2	$[\lg n-1]$
$e)$	$\sqrt{n}$	2	
$f)$	$\sqrt{n}$	1	
$g)$	$n^{1/3}$	2	
$h)$	$n/\lg n$	2	