$f = \Omega(g) \Rightarrow f(n) \geq C \cdot g(n)$, $n > K$ also proves $f = O(g)$
$f = O(g) \Rightarrow f(n) \leq C \cdot g(n)$, $n > K$
$f = \Theta(g) \Rightarrow f = \Omega(g)$ and $f = O(g)$

Chapter 0 - Exercises

0.1. In each of the following situations, indicate whether $f = O(g)$, or $f = \Omega(g)$, or both (in which case $f = \Theta(g)$).

	$f(n)$	$g(n)$
(a)	$n - 100$	$n - 200$

- $(n-100) \geq C \cdot (n-200)$, $n > 100$

when $C \geq 0$ ∴ $f = \Omega(g)$ meaning $f = O(g)$

∴ $\boxed{f = \Theta(g)}$

(b) $n^{1/2}$ $\quad$ $n^{2/3}$

- $n^{1/2} \geq 1 \cdot n^{2/3}$, $n > 1$ $\quad$ $f \neq \Omega(g)$

$n^{1/2}$ will never be greater than $C \cdot n^{2/3}$ for any value of n.

$n^{1/2} \leq C \cdot n^{2/3}$, $n > 1$

$n^{1/2}$ is always less than $C \cdot n^{2/3}$ for any value of n

∴ $\boxed{f = O(g)}$

(c) $100n + \log n$ $\quad$ $n + (\log n)^2$

- $100n + \log n \geq C \cdot (n + (\log n)^2)$, $n > 0$

Let $C = 1$

$100n + \log n \geq n + (\log n)^2$, $n > 0$ ∴ $f = \Omega(g)$

meaning $f = O(g)$ ∴ $\boxed{f = \Theta(g)}$

(d) $n \log n$ $\quad$ $10n \log 10n$

$n \log n \geq C \cdot 10n \log 10n$, $n > 1$

$n = 2$, $.60 \geq 26.02C$

$.60 / 26.02 \geq C \Rightarrow C = .023059$

∴ $f = \Omega(g)$ means $f = O(g) \Rightarrow \boxed{f = \Theta(g)}$

(e) $\log 2n$ $\quad$ $\log 3n$

$\log 2n$ is $O(\log n)$ and $\log 3n$ is $O(\log n)$

$\therefore \boxed{f = \Theta(g)}$

(f) $10\log n$ $\quad$ $\log(n^2)$

$10\log n \rightarrow \log n$ | $\log(n^2) \rightarrow 2\log n \rightarrow \log n$

$\therefore$ $10\log n$ is $O(\log n)$ and $\log(n^2)$ is $O(\log n)$

$\therefore \boxed{f = \Theta(g)}$

(g) $n^{1.01}$ $\quad$ $n\log^2 n$

	$n^{1.01}$	$n\log^2 n$
$n=10$	10.23	15.21 >
$n=100$	104.71	664.39 >
$n=1000$	1071.52	9965.84 >

$\therefore \boxed{n^{1.01} = \Omega(n\log^2 n)}$

because $n\log^2 n$ dominates $n^{1.01}$

(h) $n^2/\log n$ $\quad$ $n(\log n)^2$

Since, n^2 is higher-order than n and logarithms have minimal significance in comparison, then $\boxed{f = \Omega(g)}$

(i) $n^{0.1}$ $\quad$ $(\log n)^{10}$

n^x will always dominate logarithms $\therefore \boxed{f = \Omega(g)}$

(j) $(\log n)^{\log n}$ $\quad$ $n/\log n$

$(\log n)^{\log n} \cdot \log n$ $\quad$ $n/\cancel{\log n} \cdot \cancel{\log n}$

$(\log n)^{\log n + 1}$ $\quad$ n

$\log[(\log n)^{\log n + 1}]$ $\quad$ $\log(n)$

$(\log n + 1)\log(\log n)$ $\quad$ $\log n$

uses $g(n)$ in equation $\therefore \boxed{f = \Omega(g)}$

(k) $\sqrt{n}$ $(\log n)^3$

$n^{1/2}$ $(\log n)^3$

$n^{0.5}$ will always dominate logarithms $\therefore$ $\boxed{f = \Omega(g)}$

(l) $n^{1/2}$ $5^{\log_2 n}$

c^n (constant) dominates $\sqrt{n}$ $\therefore$ $\boxed{f = O(g)}$

(m) $n2^n$ 3^n

If the comparison was between 2^n and 3^n then 3^n would dominate 2^n but because 2^n is multiplied by a factor of n as $n2^n$ then $n2^n$ dominate 3^n $\therefore$ $\boxed{f = \Omega(g)}$

(n) 2^n 2^{n+1}

$2^n \cdot 2$ (can be simplified)

2^n

$f(n) = 2^n$ and $g(n) = 2^n$ are equal $\therefore$ $\boxed{f = \Theta(g)}$

(o) $n!$ 2^n

$\boxed{f = \Omega(g)}$ because $n!$ dominates 2^n at some point.

(p) $(\log n)^{\log n}$ $2^{(\log_2 n)^2}$

$\log[(\log n)^{\log n}]$ $\log[(2)^{(\log_2 n)^2}]$

$\log n \cdot \log(\log n)$ $(\log_2 n)^2 \log(2)$

$g(n)$ will be greater as $f(n)$ ends up taking the log of a decimal $\therefore$ $\boxed{f = O(g)}$

(q) $\sum_{i=1}^{n} i^k$ n^{k+1}

$\underbrace{1^k + 2^k + 3^k + \dots + n^k}_{\text{Constant}}$ n^{k+1}

$n \cdot n^k$

$c \cdot n^k = n^k$ $\Rightarrow n^{k+1} > n^k$ $\therefore$ $\boxed{f = O(g)}$

0.2 Show that, if c is a positive real number, then $g(n) = 1 + c + c^2 + \ldots + c^n$ is:

(a) $\Theta(1)$ if $c < 1$

Choosing c to be .5 than the series becomes

$g(n) = 1 + 0.5 + .25 + .125 + \ldots + .5^n$

$g(n)$ is decreasing $\therefore$ the sum of $\boxed{g(n) = 1 \text{ or } \Theta(1)}$

(b) $\Theta(n)$ if $c = 1$

If $c = 1$ then $g(n) = 1 + 1 + (1)^2 + \ldots + (1)^n$

then $g(n) = 1 + 1 + 1 + \ldots + 1 \Rightarrow \boxed{g(n) = n \text{ or } \Theta(n)}$

(c) $\Theta(c^n)$ if $c > 1$

Choosing c to be 2 then the series becomes

$g(n) = 1 + 2 + 2^2 + \ldots + 2^n$ then

$g(n) = 1 + 2 + 4 + \ldots + 2^n$.

The series is increasing which means the sum of the series is the last term. In our case, $g(n) = 2^n$ but with any constant this becomes $\boxed{g(n) = \Theta(c^n)}$.

0.3 The Fibonacci numbers $F_0, F_1, F_2, \ldots,$ are defined by the rule:

$$F_0 = 0,\ F_1 = 1,\ F_n = F_{n-1} + F_{n-2}$$

In this problem we will confirm that this sequence grows exponentially fast and obtain some bounds on its growth.

(a) Use induction to prove that $F_n \geq 2^{0.5n}$ for $n \geq 6$.

<u>Base Case</u> $(n = 6)$

$F_0 = 0,\ F_1 = 1,\ F_2 = F_1 + F_0 = 1,\ F_3 = F_2 + F_1 = 2$

$F_4 = F_3 + F_2 = 3,\ F_5 = F_4 + F_3 = 5,\ F_6 = F_5 + F_4 = 8$

$F_n = 2^{(0.5)(6)} = 2^3 = 8$

$\therefore$ since $F_6 = F_5 + F_4 = 8$ and $F_6 \geq 2^{(0.5)(6)} = 8$,

this case is true ✓

Case: $n = K$

$F_K = F_{K-1} + F_{K-2}$ and $F_K \geq 2^{0.5K}$

$F_{K-1} + F_{K-2} \geq 2^{0.5K}$

True for all values of K. ✓

Case: $n = K+1$

$F_{K+1} = F_K + F_{K-1}$ and $F_{K+1} \geq 2^{(0.5)(K+1)}$

$F_K + F_{K-1} \geq 2^{0.5K} + 2^{(0.5)(K-1)}$

$\vdots \quad \geq 2^{0.5K} + 2^{(0.5K - 0.5)}$

$\vdots \quad \geq 2^{0.5K} + (2^{0.5K})(2^{-0.5})$

$F_K + F_{K-1} \geq 2^{0.5K}(1 + 2^{-0.5})$

$F_{K+1} \geq 2^{0.5(K+1)} \Rightarrow F_{K+1} \geq (2^{0.5K})(2^{0.5})$

Since $(1 + 2^{-0.5}) = 1.707106781 > 2^{0.5} = 1.414213562$

$\therefore$ the case $n = K+1$ is also true and the equations hold.

(b)
(c) Find a constant $c < 1$ such that $F_n \leq 2^{cn}$ for all $n \geq 0$. Show that your answer is correct.

$F_{n-1} + F_{n-2} \leq 2^{cn}$

$2^{c(n-1)} + 2^{c(n-2)} \leq 2^{cn}$

$2^{c(n-1)-c+c} + 2^{c(n-2)} \leq 2^{cn}$

$2^{cn-c-c+c} + 2^{c(n-2)} \leq 2^{cn}$

$2^{cn-2c+c} + 2^{c(n-2)} \leq 2^{cn}$

$2^{c(n-2)+c} + 2^{c(n-2)} \leq 2^{cn}$

$(2^{c(n-2)})2^c + 2^{c(n-2)} \leq 2^{cn}$

$2^{c(n-2)}(2^c + 1) \leq 2^{cn}$

Since, $F_0 = 0$, and $F_1 = 1$ we start at $n = 2$

$F_2 \leq 2^{2c}$

$\therefore \quad 2^c + 1 \leq 2^{2c}$, Let $2^c = x$

$x + 1 \leq (2^c)^2 \Rightarrow \overset{0}{\cancel{x+1}} \leq x^2 - x - 1$

$x = \frac{1 \pm \sqrt{1+4}}{2} = \frac{1 \pm \sqrt{5}}{2} \Rightarrow 2^c \geq \frac{1 \pm \sqrt{5}}{2}$

$\cancel{\log_2}(2^c) \geq \log_2\left(\frac{1 \pm \sqrt{5}}{2}\right) \Rightarrow \boxed{c \geq 1.618}$ ← also satisfies Ω

0.4 Is there a faster way to compute the nth Fibonacci number than by fib2(page 13)? One idea involves matrices.

We start by writing the equations $F_1 = F_1$ and $F_2 = F_0 + F_1$ in matrix notation

$$\begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$$

Similarly,

$$\begin{pmatrix} F_2 \\ F_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$$

and in general

$$\begin{pmatrix} F_n \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n\begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$$

So, in order to compute F_n, it suffices to raise this 2×2 matrix, call it X, to the nth power.

(a) Show that two 2×2 matrices can be multiplied using 4 additions and 8 multiplications.

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \Rightarrow \begin{bmatrix} (1)(1)+(1)(1) & (1)(1)+(1)(1) \\ (1)(1)+(1)(1) & (1)(1)+(1)(1) \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

$\therefore$ Multiplications = 8, Additions = 4

But how many matrix multiplications does it take to compute X^n

(b) Show that $O(\log n)$ matrix multiplications suffice for computing X^n.

(Hint: Think about computing X^8)

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ 3 & 5 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 5 \\ 5 & 8 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 5 \\ 5 & 8 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 5 & 8 \\ 8 & 13 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 8 & 13 \\ 13 & 21 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 13 & 21 \\ 21 & 34 \end{pmatrix}$$

or $X^8 = X \cdot X \cdot X \cdot X \cdot X \cdot X \cdot X \cdot X = X^2 \cdot X^2 \cdot X^2 \cdot X^2$

$= X^4 \cdot X^4$ or 3 matrix multiplication steps

and $\log_2(8) = 3$ $\therefore$ fib3 runs at $O(\log n)$.

(c) Show that all intermediate results of fib3 are $O(n)$ bits long

By induction:

Base Case $(n=1)$ = 1 bit

$F_1 = 1$ $\therefore$ Base case satisfied and satisfies $n = k$.

000, 001, 010, 011, 100, 101, 110

$$\begin{array}{r} 11 \\ +\ 11 \\ \hline 110 \end{array}$$

Case 2: $n = k+1$

3 + 3 in binary is 11 + 11 = 110 or n+1 bits

$\therefore$ case where $n = k+1$ is satisfied for all Fibonacci numbers and $O(n)$ is the worst case of fib3.

(d) Let $M(n)$ be the running time of an algorithm for multiplying n-bit numbers, and assume that $M(n) = O(n^2)$ (the school method for multiplication, recalled in Chapter 1, achieves this). Prove that the running time of fib3 is $O(M(n)\log n)$.

$M(n)$ is the running time for multiplying n-bit integers

$\log n$ is the running time for fib3

If there are $M(n)$ multiplications of integers in fib3 then the total running time is $O(M(n)\log n)$.

(e) Can you prove that the running time of fib3 is $O(M(n))$? (Hint: The lengths of the numbers being multiplied get doubled with every squaring.)

Integers $= 1, 2, 3, \ldots, n$

1 $g(n) = M(1) + M(2) + M(3) + \ldots + M(n)$; $M(n) = n^2$

2 $g(n) = (1)^2 + (2)^2 + (3)^2 + \ldots + n^2$

3 $g(n) = 1 + 4 + 9 + \ldots + n^2$ ← increasing geometric series

∴ the sum of the series is the last term.

From line 1, we can see the final term is $M(n)$

∴ the running time of fib3 is $\boxed{O(M(n))}$

Finally, there is a formula for the Fibonacci numbers:

$$F_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$