$f(n) = O(g(n)) \iff \exists\, c > 0$ e $n_0 \geq 0$ $(f(n) \leq c\,g(n),\ p/\ n \geq n_0) \iff \lim_{n \to +\infty} \frac{f(n)}{g(n)} < +\infty$ (c não pode crescer c/ n

$f(n) \in \Theta(g(n)) \iff f(n) \in O(g(n))$ E $f(n) \in \Omega(g(n))$ c ñ pode diminuir c/ n

## Exercises

$f(n) \in \Omega(g(n)) \iff \exists\, c > 0$ e $n_0 \geq 0$ $(f(n) \geq c\,g(n)) \iff \lim_{n \to +\infty} \frac{f(n)}{g(n)} > 0$

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) | $n - 100$ | $n - 200$ | $f(n) = \Theta(g(n))$ pois $n$ é o termo de maior ordem |
| O e ñ Ω | (b) | $n^{1/2}$ | $n^{2/3}$ | |
| | (c) | $100n + \log n$ | $n + (\log n)^2$ | Θ, pois $n$ é o termo de maior ordem |
| Θ | (d) | $n \log n$ | $10n \log 10n$ $= 10\,n \log n + 10n \log 10$ | |
| Θ | (e) | $\log 2n$ | $\log 3n$ | $\log 2n = \log n + \log 2$; $\log 3n = \log n + \log 3$ |
| Θ | (f) | $10 \log n$ | $\log(n^2)$ $= 2 \log n$ | |
| Ω | (g) | $n^{1.01}$ | $n \log^2 n$ | |
| Ω | (h) | $n^2 / \log n$ | $n(\log n)^2$ | " > " $n^{2-\epsilon}$; " ≤ " $n^{1+\epsilon}$, p/ $\epsilon > 0$ mas $\epsilon << 1$ |
| Ω | (i) | $n^{0.1}$ | $(\log n)^{10}$ | derivar a razão 10x |
| Ω | (j) | $(\log n)^{\log n}$ | $n / \log n$ | $(\log n)^{\log n} \geq b^{\log_b n} = n > n/\log n$ |
| Ω | (k) | $\sqrt{n}$ $= n^{1/2}$ | $(\log n)^3$ | |
| O | (l) | $n^{1/2}$ | $5^{\log_2 n}$ | $> 4^{\log_2 n} = 2^{2\log_2 n} = (2^{\log_2 n})^2 = n^2$ |
| O | (m) | $n2^n$ | $3^n$ | exponencial c/ maior base cresce mais rápido |
| Θ | (n) | $2^n$ | $2^{n+1}$ $= 2 \cdot 2^n$ | |
| Ω | (o) | $n!$ | $2^n$ | |
| Ω | (p) | $(\log n)^{\log n}$ | $2^{(\log_2 n)^2}$ | $= 2^{\log_2 n \cdot \log_2 n} = n^{\log_2 n}$ |
| Θ | (q) | $\sum_{i=1}^{n} i^k$ | $n^{k+1}$ | $= n^k + (n-1)^k + (n-2)^k + \dots + 1^k \leq n \cdot n^k = n^{k+1}$; $\geq \frac{n}{2}$ |

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

(a) $\Theta(1)$ if $c < 1$.
(b) $\Theta(n)$ if $c = 1$.
(c) $\Theta(c^n)$ if $c > 1$.

The moral: in big-$\Theta$ terms, the sum of a geometric series is simply the first term if the series is strictly decreasing, the last term if the series is strictly increasing, or the number of terms if the series is unchanging.

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 $n \geq 6$.
(b) Find a constant $c < 1$ such that $F_n \leq 2^{cn}$ for all $n \geq 0$. Show that your answer is correct.
(c) What is the largest $c$ you can find for which $F_n = \Omega(2^{cn})$?

0.4. Is there a faster way to compute the $n$th 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:

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

Similarly,

$$\begin{pmatrix} F_2 \\ F_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^2 \cdot \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$$

Q.1)

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

$$\boxed{n^{1/2} = n^{3/6} \underset{n>1}{\leq} n^{3/6} \cdot n^{1/6} = n^{4/6} = \overset{c}{1}\, n^{2/3}}$$

$f(n) \leq c\, g(n)$ p/ $c = 1$ e $n_0 = 1 \Rightarrow f(n) \in O(g(n))$

Por absurdo $f(n) \geq c\, g(n)$ p/ $n \geq n_0$

$$n^{1/2} \geq c \cdot n^{2/3} \Rightarrow \boxed{c \leq \frac{n^{1/2}}{n^{2/3}} = \frac{n^{3/6}}{n^{4/6}} = \frac{\cancel{n^{3/6}}}{n^{1/6} \cdot \cancel{n^{3/6}}} = 1/n^{1/6}}$$

como $c \leq 1/n^{1/6}$ p/ $n \geq n_0 \Rightarrow \nexists c > 0$, logo $f(n) \notin \Omega(g(n))$

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

$$100n + \log n \underset{n \geq 0}{\leq} 101n \underset{n \geq 0}{\leq} 101(n + (\log n)^2)$$

$f(n) \leq c\,g(n)$, p/ $c = 101$ e $n_0 = 0 \Rightarrow f(n) \in O(g(n))$

| $n$ | $\log_2 n$ | $(\log n)^2$ |
|---|---|---|
| 1 | 0 | 0 |
| 2 | 1 | 1 |
| 4 | 2 | 4 |
| 8 | 3 | 9 |
| 16 | 4 | 16 |
| 32 | 5 | 25 |
| 64 | 6 | 36 |

$$100n + \log n \geq 2n \underset{n \geq 32}{\geq} \overset{c}{1}(n + (\log n)^2)$$

$f(n) \geq c\, g(n)$, p/ $c = 1$ e $n_0 = 32 \Rightarrow f(n) \in \Omega(g(n))$

g) $f(n) = n^{1+\epsilon}$, p/ $\epsilon > 0$ $\quad g(n) = n \log^2 n$

$$f(n) \in \Omega(g(n)) \Leftrightarrow \exists c > 0, n_0 \geq 0 \ (f(n) \geq c\,g(n), \text{ p/ } n \geq n_0) \Leftrightarrow \lim_{n \to +\infty} \frac{f(n)}{g(n)} > 0$$

$$\boxed{\lim_{n \to +\infty} \frac{n^{1+\epsilon}}{\cancel{n} \log^2 n}} = \lim_{n \to +\infty} \frac{n^{\epsilon}}{\log^2 n} \overset{\text{derivando}}{=} \lim_{n \to +\infty} \frac{\epsilon \cdot n^{\epsilon - 1}}{(2/\ln b) \cdot \frac{\log n}{n}}$$

$$\boxed{\frac{d\, n^{\epsilon}}{dn} = \epsilon \cdot n^{\epsilon - 1}}$$

$$\boxed{\frac{d \log^2 n}{dn}} = \frac{d \log^2 n}{d \log n} \cdot \frac{d \log_b n}{dn} \quad \left(\log_b n = \frac{\ln n}{\ln b}\right)$$

$$= 2 \cdot \log n \cdot \frac{1}{\ln b\ n} = \boxed{\frac{2}{\ln b} \cdot \frac{\log n}{n}}$$

$$= \frac{\epsilon \ln b}{2} \lim_{n \to +\infty} \frac{n^{\epsilon} \cdot \cancel{n^{-1}} \cdot \cancel{n}}{\log_b n}$$

derivando

$$= \frac{\epsilon \ln b}{2} \lim_{n \to +\infty} \frac{\epsilon \cdot n^{\epsilon - 1}}{\ln b \cdot (1/n)} \simeq \lim_{n \to +\infty} n^{\epsilon} \cdot \cancel{n^{-1}} \cdot \cancel{n} = \boxed{+\infty}$$

Como $+\infty > 0$ Temos $f(n) \in \Omega(g(n))$

Como $+\infty > c$ p/ qualquer constante $c$ então $f(n) \notin O(g(n))$

Note que mostramos que qualquer polinômio de expoente positivo cresce assintoticamente mais rápido que a função $\log n$

m) $f(n) = n \cdot 2^n$ $\quad g(n) = 3^n$

$$n \cdot 2^n = n \cdot \underbrace{2 \cdot 2 \cdot 2 \cdot \ldots \cdot 2}_{n \text{ vezes}} \qquad 3^n = \underbrace{3 \cdot 3 \cdot 3 \cdot \ldots \cdot 3}_{\text{vezes}} = \left(2 \cdot \frac{3}{2}\right)^n = 2^n \cdot \left(\frac{3}{2}\right)^n$$

$$f(n) \in O(g(n)) \Leftrightarrow \lim_{n \to +\infty} \frac{f(n)}{g(n)} \leq c < +\infty$$

$$\boxed{\lim_{n \to +\infty} \frac{n \cdot 2^n}{3^n}} = \lim_{n \to +\infty} \frac{n \cdot \cancel{2^n}}{\cancel{2^n} \cdot (3/2)^n} = \lim_{n \to +\infty} \frac{n}{(3/2)^n} \overset{\text{derivando}}{=} \lim_{n \to +\infty} \frac{1}{\ln(3/2) \cdot (3/2)^n}$$

$$\boxed{= 0}$$

$$\boxed{\frac{dn}{dn} = 1}$$

$$\boxed{\frac{d(3/2)^n}{dn}} = \frac{d\left(e^{\ln 3/2}\right)^n}{dn} = \frac{d\, e^{n \cdot \ln 3/2}}{d\, n \cdot \ln 3/2} \cdot \frac{d\, n \cdot \ln 3/2}{dn}$$

$$= e^{n \cdot \ln 3/2} \cdot \ln 3/2 = \boxed{\ln(3/2) \cdot (3/2)^n}$$

$$\lim_{n \to +\infty} \frac{f(n)}{g(n)} = 0 < c \Rightarrow f(n) \in O(g(n))$$

$$\lim_{n \to +\infty} \frac{f(n)}{g(n)} = 0 < c \Rightarrow f(n) \notin \Omega(g(n))$$

o) $f(n) = n!$ $\quad g(n) = 2^n$

$$\boxed{n! = \underbrace{n \cdot (n-1) \cdot (n-2) \ldots 3 \cdot 2}_{n-1 \text{ termos}} \cdot 1 \underset{n \geq 1}{\geq} 2^{n-1} \cdot 1 = \frac{2}{2} \cdot 2^{n-1} = \overset{c}{\frac{1}{2}} \cdot 2^n}$$

$f(n) \in \Omega(g(n))$ p/ $c = \frac{1}{2}$ e $n_0 = 1$

Por absurdo, supor que $\exists c > 0$ e $n_0 \geq 0$:

$$n! \leq c \cdot 2^n \Rightarrow \cancel{2^n} \cdot c \geq n! = \underbrace{n \cdot (n-1) \cdot (n-2) \ldots}_{n-3 \text{ termos}} 4 \cdot 3 \cdot 2 \cdot 1 \geq 4^{n-3} \cdot 3 \cdot 2 \cdot 1$$

$$\geq 2^{2n-6} = \cancel{2^n} \cdot 2^{n-6} \Rightarrow c \geq 2^{n-6}, \text{ p/ } n \geq n_0,$$

logo $c$ ñ é constante (Contradição). Portanto, $f(n) \notin O(g(n))$

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

$$\boxed{\sum_{i=1}^{n} i^k} = \underbrace{n^k + (n-1)^k + (n-2)^k + (n-3)^k + \cdots + 3^k + 2^k + 1^k}_{n \text{ termos}}$$

$$\leq n^k + n^k + n^k + n^k + \ldots + n^k + n^k + n^k = n \cdot n^k = \boxed{\overset{c}{1} \cdot n^{k+1}}$$

$f(n) \in O(g(n))$ p/ $c = 1$ e $n_0 = 1$

$$\boxed{\sum_{i=1}^{n} i^k} = \underbrace{n^k + (n-1)^k + (n-2)^k + \cdots + \left(n - \frac{n}{2} + 1\right)^k}_{n/2 \text{ termos} > \text{ que } n/2} + \left(n - \frac{n}{2}\right)^k + \ldots + 2^k + 1^k$$

$$\geq \left(\frac{n}{2}\right)^k + \left(\frac{n}{2}\right)^k + \left(\frac{n}{2}\right)^k + \cdots + \left(\frac{n}{2}\right)^k = \frac{n}{2} \cdot \frac{n^k}{2^k} = \boxed{\overset{c}{\left(\frac{1}{2^{k+1}}\right)} n^{k+1}}$$

$f(n) \in \Omega(g(n))$ p/ $c = \frac{1}{2^{k+1}}$ e $n_0 = 2$

and in general

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

So, in order to compute $F_n$, it suffices to raise this $2 \times 2$ matrix, call it $X$, to the $n$th power.

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

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 $X^8$.)

Thus the number of arithmetic operations needed by our matrix-based algorithm, call it `fib3`, is just $O(\log n)$, as compared to $O(n)$ for `fib2`. *Have we broken another exponential barrier?*

The catch is that our new algorithm involves multiplication, not just addition; and multiplications of large numbers are slower than additions. We have already seen that, when the complexity of arithmetic operations is taken into account, the running time of `fib2` becomes $O(n^2)$.

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

(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 $O(M(n) \log n)$.

(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.)

In conclusion, whether `fib3` is faster than `fib2` depends on whether we can multiply $n$-bit integers faster than $O(n^2)$. Do you think this is possible? (The answer is in Chapter 2.)

Finally, there is a formula for the Fibonacci numbers:

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

So, it would appear that we only need to raise a couple of numbers to the $n$th power in order to compute $F_n$. The problem is that these numbers are irrational, and computing them to sufficient accuracy is nontrivial. In fact, our matrix method `fib3` can be seen as a roundabout way of raising these irrational numbers to the $n$th power. If you know your linear algebra, you should see why. (*Hint:* What are the eigenvalues of the matrix $X$?)