	Departamento de Matemática da Universidade de Coimbra		
	Análise de Algoritmos	**Micro-teste 1**	**4/03/2009**

1. Os números de Fibonacci, $F_0, F_1, F_2, \ldots$, são definidos pela regra

$$F_n = \begin{cases} 0 & \text{se } n = 0 \\ 1 & \text{se } n = 1 \\ F_{n-1} + F_{n-2} & \text{se } n > 1 \end{cases}$$

Com este problema vamos confirmar que esta sequência cresce a um ritmo exponencial e obter alguns limites para o seu crescimento.

(a) Usar o Método de Indução para provar que $F_n \geq 2^{0,5\,n}$ se $n \geq 6$.

(b) Encontre uma constante $c < 1$ tal que $F_n \leq 2^{cn}\,, \forall n \geq 0$. Prove a correcção deste resultado.

(c) Qual é o maior c que consegue encontrar tal que $F_n = \Theta(2^{cn})$.

2. Mostre que, para quaisquer constantes reais, a e b com $b > 0$, se tem

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

3. Verifique se

(a) $2^{n+1} = \mathcal{O}(2^n)$?

(b) $2^{2n} = \mathcal{O}(2^n)$?

4. Existirá um modo mais rápido de calcular o n–ésimo número de Fibonacci do que o algoritmo *fiboP*?

Uma hipótese parece consistir no uso de matrizes: Se escrevermos as equações $F_1 = F_1$ e $F_2 = F_0 + F_1$ na sua forma matricial obtemos:

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

Ora,

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

Em geral, ter-se-á;

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

Portanto, para calcular F_n, basta calcular a matriz dos zeros e uns, seja ela X, à n–ésima potência.

(a) Mostre que, para calcular o produto de duas matrizes 2×2, bastam 4 somas e 8 multiplicações.

(b) Mas, em geral, quantas multiplicações de matrizes são necessárias para calcular X^n?

(c) Mostre que bastam $\mathcal{O}(\log n)$ multiplicações de matrizes para calcular X^n.
Sugestão: Pense no modo de calcular X^8.

(c) Este algoritmo baixa a barreira linear do pior dos casos anteriormente encontrada para o cálculo do F_n. Comente esta afirmação (em não mais do que 4 linhas).