Complexité des algorithmes

Exercice 1

On considère la fonction suivante :

```
G(int n, double x, double y) = if n=0 then y
                               else if n mod 2 = 0 then G(n/2, x*x, y)
                               else G(n-1, x, y * x)
```

1. Montrer que si $n \geq 0$, alors `G(`n`,`x`,`y`)` calcule $y \times x^n$.
2. À quoi peut bien servir cette fonction ?
3. Soient n et p deux entiers naturels tels que $2^p \leq n < 2^{p+1}$. Montrer que le calcul de $G(n, x, y)$ demande un nombre d'appels de G compris entre $p+2$ et $2p+2$. (Par exemple, le calcul de $G(6, x, 1)$ se fait avec les valeurs suivantes de n : 6, 3, 2, 1, 0 (soit 5 appels de `G`)).

Exercice 2

Pour chacune des paires de fonctions suivantes $f(n)$ et $g(n)$, dire si f est $O(g)$, si f est $\Omega(g)$ et si f est $\Theta(g)$:

a) $f(n) = n^{10}$; $g(n) = 2^{n/2}$
b) $f(n) = n^{3/2}$; $g(n) = n\log^2(n)$
c) $f(n) = \log(n^3)$; $g(n) = \log(n)$
d) $f(n) = \log(3^n)$; $g(n) = \log(2^n)$
e) $f(n) = 2^n$; $g(n) = 2^{n/2}$
f) $f(n) = n^2$; $g(n) = (n/2)^2$

Exercice 3

Soient $f(n)$ et $g(n)$ des fonctions positives. À quelle condition (nécessaire et suffisante) a-t-on que $\Theta(f(n) + g(n)) = \Theta(f(n))$? Justifier.

Exercice 4

Démontrer que pour deux constantes réelles a et b quelconques, avec $b > 0$, $\Theta((n+a)^b) = \Theta(n^b)$.

Exercice 5

On suppose que $f(n)$ est $\Theta(1 + f(\frac{n}{2}))$. Peut-on en déduire que $f(n)$ est $\Theta(\log n)$?

Exercice 6

Soit F_n une fonction de Fibonacci. Déduire de la relation

$$\begin{bmatrix} F_{n+2} \\ F_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix}$$

un algorithme récursif qui calcule F_n en temps $O(\log n)$.

Exercice 7

Soit u_n la fonction de Fibonacci, avec $u_0 = u_1 = 1$. Déduire des égalités suivantes (avec $n > 0$)

$$\begin{aligned} u_{2n} &= u_n^2 + u_{n-1}^2 \\ u_{2n+1} &= u_n^2 + 2u_n u_{n-1} \\ u_{2n+2} &= 2u_n^2 + 2u_n u_{n-1} + u_{n-1}^2 \end{aligned}$$

un algorithme récursif qui calcule u_n en temps $O(\log n)$.

Question subsidiaire : prouver les égalités ci-dessus.