CORRIGE + BAREME PARTIEL : Algorithme et Programmation Université Paris-Dauphine MIDO L2

Exercice 1 (2 points) *Combien vaut x ?*

$$\left(\sqrt{\frac{2}{5}}\right)^{7x+1} = \frac{125}{8}$$

$x = -1$ 2 points, toute autre réponse = 0

Exercice 2 (3 points) *Complétez la condition* `while` *pour que, pour tout entier n,* `iquo(n,b)` *retourne l'unique entier q tel que $n = qb + r$ avec $0 \leq r < b$.*

```
def iquo(n,b):
        q=0
        while
                q = q+1
        return q
```

Démontrez alors la validité de `iquo(n,b)`*.*

`while( n-b*q >= b )` 2 points,
toute autre réponse = 0 mais attention: `b*q+b-1<n` est équivalent par exemple...
Si juste, on regarde la preuve
Preuve: on sort de la boucle avec $r := n - bq < b$ et $n - b(q-1) \geq b$ d'où $r = n - bq \geq 0$.
1 point

Soient a, b des constantes entières ≥ 2 et n une variable entière (dans tous les exercices suivants).

Exercice 3 (2 points) *Montrez que*

1. $\log_b n = \log_b a \times \log_a n$.
2. $a^{\log_b n} = n^{\log_b a}$.

1. On pose $b^u = n$, $b^v = a$ et $a^w = n$.
Ainsi $b^u = n = a^w = (b^v)^w = b^{vw}$, et donc $u = vw$ ce qui implique l'énoncé 1 point
2. De plus, $a^u = (b^v)^u = b^{uv} = (b^u)^v = n^v$, ce qui implique l'énoncé 1 point

Soient $f(n)$ et $g(n)$ des fonctions entières croissantes, positives et tendant vers $+\infty$.

Exercice 4 (2.5 points) *Montrez que*

1. $f(n) = \Theta(g(n))$ *si et seulement si* $g(n) = \Theta(f(n))$.
2. $f(n) = \Theta(h(n))$ *et* $f(n) = \Theta(h(n))$ *seulement si* $f(n) + g(n) = \Theta(h(n))$.
3. $f(n) = \Theta(h(n))$ *et* $f(n) = \Theta(h(n))$ *seulement si* $f(n)g(n) = \Theta(h^2(n))$.
4. $af(n) + b = \Theta(g(n))$ *si et seulement si* $f(n) = \Theta(g(n))$.

1. $\exists c, d, n_o > 0$ telles que $cg(n) \leq f(n) \leq dg(n)$, $\forall n \geq n_o$, implique $\frac{1}{d} f(n) \leq g(n) \leq \frac{1}{c} f(n)$, $\forall n \geq n_o$. De même pour la réciproque. 0.5 point
2. (il faut lire bien sûr : $f(n) = \Theta(h(n))$ et $g(n) = \Theta(h(n))$ seulement si $f(n) + g(n) = \Theta(h(n))$.)
$\exists c, d, n_o > 0$ telles que $ch(n) \leq f(n) \leq dh(n)$, $\forall n \geq n_o$ et $\exists c', d', n'_o > 0$ telles que $c'h(n) \leq g(n) \leq d'h(n)$, $\forall n \geq n'_o$ implique

$\min\{c, c'\}h(n) \leq f(n) + g(n) \leq \max\{d, d'\}h(n)$, $\forall n \geq \max\{n_o, n'_o\}$. 0.5 point
et
3. $cc'h^2(n) \leq f(n)g(n) \leq dd'h^2(n)$, $\forall n \geq \max\{n_o, n'_o\}$. 0.5 point
4. Par (1), il suffit de montrer que $af(n) + b = \Theta(f(n))$.
Trivialement, on a: $af(n) = \Theta(f(n))$. (prendre $c = d = a$ dans la def.)
Il suffit donc de montrer que $f(n) + b = \Theta(f(n))$.

Les hypothèses sur $f(n)$ impliquent $\exists n_o$ tel que $f(n) \geq b$, $\forall n \geq n_o$; c'est-à-dire $f(n) \leq f(n) + b \leq 2f(n)$, ce qui implique le résultat. (prendre $c = 1$ et $d = 2$ dans la def.) 1 point

Exercice 5 (3 points) *Montrez que*

1. $\sum_{i=1}^{i=n} \frac{1}{i} = \Theta(\log n)$.
2. $\log(n!) = \Theta(n \log n)$.

1. Il existe un entier p tel que $2^p \leq n < 2^{p+1}$, on a alors

$$\frac{p}{2} = \sum_{i=1}^{p} 2^{i-1}2^{-i} = \frac{1}{2^p} + \ldots + \frac{1}{8} + \frac{1}{4} + \frac{1}{4} + \frac{1}{2} \leq \sum_{i=1}^{n} \frac{1}{i} \leq \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \ldots + \frac{1}{2^p} = \sum_{i=0}^{p} 2^i 2^{-i} = p + 1$$

Par l'exercice 4.4, on a $\frac{p}{2}$ et $p + 1 = \Theta(p) = \Theta(\log n)$; ce qui implique le résultat. 1.5 points

2. On a $\frac{n}{2}^{\frac{n}{2}} \leq n! \leq n^n$, il suffit donc de montrer que $\frac{n}{2}\log(\frac{n}{2}) = \Omega(n \log n)$, et, d'après Exercice 4.4, il suffit donc de montrer que $\log n - \log 2 = \Omega(\log n)$. Pour cela, il suffit de montrer que $f(n) - b = \Theta(f(n))$ (puisque $\log n$ satisfait les conditions de $f(n)$). Les hypothèses sur $f(n)$ impliquent $\exists n_o$ tel que $\frac{1}{2}f(n) \geq b$, $\forall n \geq n_o$; c'est-à-dire $\frac{1}{2}f(n) \leq f(n) - b \leq f(n)$, ce qui implique le résultat. (prendre $c = \frac{1}{2}$ et $d = 1$ dans la def.) 1.5 points

Soit c une constante entière.

Exercice 6 (3 points) *Montrez que*

1. $\sum_{i=1}^{i=n} i^c = \Theta(n^{c+1})$.
2. $(n + (-1)^a b)^c = \Theta(n^c)$.

1. On a $\sum_{i=1}^{i=n} i^c \leq n\, n^c = O(n^{c+1})$. De plus,

$$\sum_{i=1}^{n} i^c \geq \sum_{i=0}^{\lfloor n/2 \rfloor} \left(\frac{n}{2} + i\right)^c = \frac{1}{2^c} \sum_{i=0}^{\lfloor n/2 \rfloor} (n + 2i)^c \geq \frac{1}{2^{c+1}} n^{c+1} = \Omega(n^{c+1})$$

1.5 points

2. Par Exercice 4.3, il suffit de montrer (ce qui a déjà été fait pour la fin de l'Exercice 5.2) que $n + (-1)^a b = \Theta(n)$. (En effet, Exercice 4.3 implique $f(n) = \Theta(h(n))$ seulement si $f^c(n) = \Theta(h^c(n))$.) 1.5 points

Exercice 7 (2 points) *Quelle est la complexité de l'algorithme* `r(n)` *? (démontrez votre réponse)*

```
def r(n):
        global y
        for i in range(1,2*n):
                for j in range(n//i):
                        y=y+1
```

On a $\sum_{i=1}^{2n-1} \lfloor \frac{n}{i} \rfloor = n\Theta(\sum_{i=1}^{2n} \frac{1}{i}) = \Theta(n \log 2n)$. De plus, $\log 2n = \Theta(\log n)$. Donc la complexité est $\Theta(n \log n)$.

Exercice 8 (2.5 points) *Quelle est la complexité de* `p(n,c)` *? (c est une constante) (démontrez votre réponse)*

```
def p(n,i):
        if i>=1:
                for j in range(n//2):
                        p(n,i-1)
```

La complexité est $\Theta(\sum_{i=0}^{c} \left\lfloor \frac{n}{2} \right\rfloor^{c-i} = \sum_{i=0}^{c} r^i)$ avec $r = \left\lfloor \frac{n}{2} \right\rfloor$. Pour $r > 1$, on a $r^c \leq \sum_{i=0}^{c} r^i \leq (c+1)r^c$ donc $\sum_{i=0}^{c} r^i = \Theta(r^c)$. Donc la complexité est $\Theta(n^c)$.