L2 informatique - Année 2020–2021

TD d'Éléments d'Algorithmique n° 4

*** Les exercices marqués d'une étoile sont à faire à la maison.**

Exercice 1. *Mise en jambes.*
On considère la suite définie par

$$\begin{aligned} a_0 &= 0; \\ a_n &= 3 + a_{n-1} \qquad (n > 0). \end{aligned}$$

1. Donnez une forme close pour a_n.
2. Écrivez une fonction récursive qui calcule a_n en appliquant directement la définition ci-dessus. Cette fonction est-elle récursive terminale ?
3. Écrivez une fonction récursive terminale qui calcule la même chose que la fonction précédente.
4. Écrivez une fonction itérative (avec des boucles) qui calcule encore la même chose.

Exercice 2. *Récursivité et état.*
On suppose donnée une fonction `char` qui convertit un entier compris entre 0 et 9 en un caractère compris entre `'0'` et `'9'`, et une fonction `afficher` qui affiche un caractère. On note `div` le quotient de la division entière, et `mod` le reste de la division entière. On considère ensuite la fonction H définie comme suit :

```
H(n):
if n > 0 {
    H(n div 10)
    afficher(char(n mod 10))
}
```

1. Décrivez l'arbre des appels effectués lors de l'évaluation de $H(123)$.
2. Que fait cette fonction ? Est-elle récursive terminale ?
3. Écrivez une version itérative (non récursive) de cette fonction. De combien d'espace supplémentaire cette version a-t-elle besoin ? Comparez cela à la version récursive. Quel est l'interêt de la récursion ?

Exercice 3. *Suite de Fibonacci*.*
Leonardo di Pisa, dit Fibonacci, élève des lapins. Les lapins de Fibonacci ont un cycle de vie un peu particulier : sa première année, un lapin est immature, et ne se reproduit donc pas. À partir de sa deuxième année, un lapin produit chaque année un autre lapin par parthénogénèse. Les lapins sont immortels. Il y a donc un lapin l'année zéro, un lapin la première année, deux lapins la deuxième année, trois lapins la troisième année, etc.
La *suite de Fibonacci* (f_n) est définie par la récurrence suivante :

$$\begin{aligned} f_0 &= 1; \\ f_1 &= 1; \\ f_n &= f_{n-2} + f_{n-1} \qquad (n \geq 2). \end{aligned}$$

1. Écrivez l'algorithme récursif qui calcule f_n suggéré par le système d'équations ci-dessus.

2. Écrivez l'arbre des appels effectués pour calculer f_4, et comptez le nombre d'additions.
3. À quelle formule de récurrence obéit le nombre d'additions ? Donnez les premiers termes de la suite.
4. Écrivez un algorithme itératif en espace constant (sans utiliser de tableaux) qui calcule f_n. Combien d'additions fait-il pour calculer f_4 ? Pour calculer f_n ?
5. Lorsqu'on écrit une fonction sur la forme récursive terminale, on stocke en général plus d'informations dans les paramètres.

   Ecrivez un algorithme récursif terminal qui calcule f_n.
6. Implémentez vos algorithme des questions 1 et 5 (en JAVA par exemple). Comparez le temps d'exécution qu'il prend pour calculer $f_{30}, f_{35}, f_{40}, f_{45} \ldots$ sur ces deux implémentations.