

TD3 : récursivité**Exercice 1.**

- (1) Donner un algorithme récursif qui affiche les entiers de 1 à n .
- (2) Même chose de n à 1.
- (3) Puis-je modifier ces algorithmes pour qu'ils rajoutent un caractère final de retour à la ligne à la fin de la liste ?

Exercice 2.

Pour chacun des algorithmes suivant :

- (1) Déterminer pour quelles valeurs d'entrée l'algorithme termine
- (2) Calculer un exemple à la main
- (3) Expliquer ce que calcule l'algorithme

```
Fonction1
Input : un entier n
Processus :
  Si n == 0 :
    Retourner 1
  Retourner Fonction1(n+1)
```

```
Fonction2
Input : un entier n
Processus :
  Si n == 0 :
    Retourner 0
  Retourner Fonction2(n-1)+n
```

```
Fonction3
Input : un entier n
Processus :
  Si n == 0 :
    Retourner 0
  Retourner Fonction3(n-1)-n
```

```
Fonction4
Input : un entier n
Processus :
  Si n == 0 :
    Retourner 0
  Si n < 0 :
    Retourner n + Fonction4(-n)
  Retourner n + Fonction4(-n+1)
```

```
Fonction5
Input : un entier n
Processus :
  Si n <= 1 :
    Retourner 0
  Retourner 1 + Fonction5(n-2)
```

Exercice 3.

La fonction de fibonacci est définie par :

$$U_0 = 1$$

$$U_1 = 1$$

$$U_n = U_{n-1} + U_{n-2} \text{ si } n \geq 2.$$

Ces premières valeurs sont donc : 1, 1, 2, 3, 5, 8, 13, ...

Donner deux algorithmes, un itératif et un récursif, qui calculent la valeur n de la suite.

Exercice 4.

On rappelle sur un exemple le principe de l'algorithme d'Euclide du calcul du pgcd (plus grand diviseur commun).

Calcul du pgcd de 2145 et 630 : On commence par effectuer la division euclidienne de 2145 par 630

$$2145 = 630 * 3 + 255$$

puis on effectue la division de 630 par le reste obtenu 255

$$630 = 255 * 2 + 120$$

On continue jusqu'à ce que l'on trouve un reste nul.

$$255 = 120 * 2 + 15$$

$$120 = 15 * 8$$

Le pgcd est le dernier reste non nul, c'est-à-dire **15**.

Donner un algorithme récursif qui calcule le pgcd de deux nombres.

Exercice 5.

Reprendre l'algorithme de recherche dichotomique dans un tableau trié vu dans le TD1 et donner une version récursive.

Exercice 6.

Donner les complexités des algorithmes des exercices 3 et 5.

Exercice 7.

Il existe deux définitions récursives de la fonction puissance :

$$a^b = \begin{cases} 1 & \text{si } b = 0 \\ a \times a^{b-1} & \text{sinon} \end{cases}$$

$$a^b = \begin{cases} 1 & \text{si } b = 0 \\ a \times a^{b-1} & \text{si } b \text{ est impair} \\ a^{\frac{b}{2}} a^{\frac{b}{2}} & \text{si } b \text{ est pair} \end{cases}$$

Pour chacune des définitions, donner l'algorithme récursif correspondant ainsi que sa complexité.