

## Chapitre 1

### Programmation récursive

#### TD 1

#### Exercices d'application

*TD d'informatique du Lycée Louis Legrand – Jean-Pierre Becirspahic*  
<http://info-llg.fr/>

#### Savoirs et compétences :

- Alg – C15 : Récursivité : avantages et inconvénients.

### Exercice 1 – Fonction 91 de McCarthy

On considère la fonction récursive suivante :

```
■ Python
def f(n) :
    if n > 100 :
        return n - 10
    return f(f(n + 11))
```

**Question** Prouver sa terminaison lorsque  $n \in \mathbb{N}$  et déterminer ce qu'elle calcule (sans utiliser l'interpréteur de commande).

**Correction** Montrons que  $n$  est un variant de boucle.

Si  $n \geq 101$ , l'algorithme se termine.

Si  $n \in [90, 100]$ , l'algorithme appelle  $f(n + 11)$ .  $n + 11$  sera supérieur à 101. et  $f(n + 11)$  retournera donc un nombre  $a$  compris entre  $[91, 101]$ .

### Exercice 2

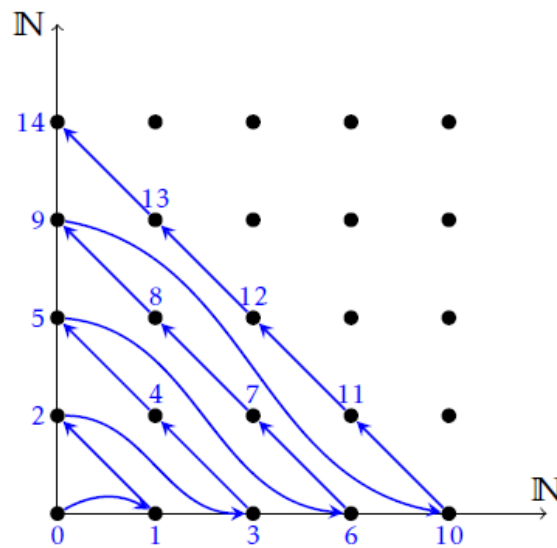
**Question 1** Écrire une fonction récursive qui calcule  $a^n$  en exploitant la relation :  $a^n = a^{n/2} \times a^{n/2}$ .

**Question 2** Écrire une fonction qui utilise de plus la remarque suivante :  $n/2 = \begin{cases} n/2 & \text{si } n \text{ est pair} \\ n/2 + 1 & \text{sinon} \end{cases}$ .

**Question 3** Effectuer le nombre de multiplications effectuées dans les deux cas.

### Exercice 3

On démontre que sur l'ensemble  $\mathbb{N} \times \mathbb{N}$  est dénombrable en numérotant chaque couple  $(x, y) \in \mathbb{N}^2$  suivant le procédé suggéré par la figure ci-dessous.



**Correction** `def polygon(*args):`

`X, Y = [], []`

`for arg in args:`

`X.append(arg[0])`

`Y.append(arg[1])`

`plt.fill(X, Y, 'b')`

`from numpy import sqrt`

`def sierpinski(n, a=(0, 0), b=(1, 0), c=(.5, sqrt(3)/2)):`

`if n == 1:`

`polygon(a, b, c)`

`else:`

`u = ((b[0]+c[0])/2, (b[1]+c[1])/2)`

`v = ((c[0]+a[0])/2, (c[1]+a[1])/2)`

`w = ((a[0]+b[0])/2, (a[1]+b[1])/2)`

`sierpinski(n-1, a, w, v)`

`sierpinski(n-1, w, b, u)`

`sierpinski(n-1, v, u, c)`