

Chapitre 1

Programmation récursive

TD 2

Exercices d'application

Savoirs et compétences :

- Alg – C15 : Récursivité : avantages et inconvénients.

Exercice 1 – Calcul de $n!$

On rappelle la définition de $n!$:

$$\forall n \in \mathbb{N} : \begin{cases} n! = \prod_{k=1}^n k & \text{si } n \geq 1 \\ n! = 1 & \text{si } n = 0 \end{cases}$$

R Pour vérifier vos résultats, vous pouvez utiliser la fonction disponible dans la bibliothèque math :

```
>>> from math import factorial
>>> print(factorial(4))
24
```

Question 1 Définir la fonction `fact_it` permettant de calculer $n!$ de façon itérative.

Question 2 Donner alors la complexité algorithmique de votre algorithme.

Question 3 Définir la fonction `fact_rec` permettant de calculer $n!$ de façon récursive.

Question 4 Évaluer $1010!$ dans les trois cas. Expliquer ce qu'il se passe ?

Exercice 2 – Suite de Fibonacci

Exercice 3 – Exponentiation rapide

Exercice 4 – Calcul de déterminant

D'après ressources de David Prévost – UPSTI

On souhaite calculer le déterminant d'une matrice (carrée) A par la formule de développement sur la première ligne (ou sur la première colonne) mais pas une autre (par exemple si on le faisait sur la troisième, la fonction ne conviendrait pas pour un déterminant 2×2).

Question 1 Écrire une fonction `extraire(A,lig,col)` qui permet de fabriquer la matrice extraite de A en rayant la ligne `lig` et la colonne `col` (dans l'optique d'obtenir son déterminant).

Question 2 Écrire une fonction récursive `determinant(A)` retournant le déterminant de A .

Question 3 Tester l'algorithme sur une matrice (de taille au moins $3,3$) et comparer le résultat à ce que retourne la fonction `det` du module `linalg` de la bibliothèque `numpy`.

Exercice 1 – Suite de Fibonacci

Exercice 2 – Recherche dichotomique

Exercice 3 – Courbe du Dragon