TP n° 14 – Synthèse algorithmique

L'objectif de ce TP est de faire une synthèse des connaissances algorithmiques acquises cette année. On a volontairement choisi de travailler sur quelques algorithmes « classiques », qui correspondent chacun à une spécification. Ce sont ces spécifications qui sont données en énoncé des exercices ci-après. Dans chaque cas, il s'agira de :

- définir un algorithme sous forme récursive ;
- proposer une implémentation ;
- vérifier sa correction ;
- déterminer sa complexité spatiale et temporelle ;
- proposer une optimisation sous forme récursive terminale.

La récursivité terminale est une forme particulière de récursivité, plus rapide et consommant moins de mémoire. S'il n'y a pas récursivité terminale, à chaque appel récursif de la fonction, le programme allouera un nouvel environnement d'exécution dans la pile, avec un risque de dépassement de la capacité de la pile.

Une fonction f est dite **récursive terminale** si, dans tous les cas, elle retourne soit une valeur fixe, soit le résultat d'un appel récursif à la fonction f (sans qu'il n'y ait aucune opération sur cette valeur).

La plupart du temps, pour transformer une fonction récursive en fonction récursive terminale, on a besoin d'introduire de nouvelles variables appelées accumulateurs ou variables auxiliaires grâce auxquelles on construit petit à petit le résultat. Lorsque la condition d'arrêt est atteinte, ce sont généralement ces variables qui sont retournées comme résultat de la fonction.

Exercice 14.1 (Échauffement)

Implémenter une fonction factorielle sous forme récursive terminale.

Les exercices qui suivent travaillent tous sur des séquences, c'est-à-dire des listes ou tableaux (objets mutables), que l'on supposera non vides.

Exercice 14.2 (Extrema)

Trouver les extremums (minimum et maximum) d'une séquence d'entiers.

❏ *On pourra commencer par une recherche du minimum.*

Exercice 14.3 (Recherche dans une séquence triée)

Trouver, le plus rapidement possible, un élément dans une séquence d'entiers triée.

Exercice 14.4 (Tri)

Trier sur place une séquence d'entiers le plus rapidement possible.

* *
*