

TD4 – Récursion

Exercice 1.*Échauffement*

La factorielle d'un entier n est l'entier $n! = 1 \times 2 \times 3 \times \dots \times n$.

-  Écrire un algorithme récursif prenant en entrée un entier n et renvoyant sa factorielle $n!$.
Quelle est la complexité de l'algorithme ?

Exercice 2.*Tableau à deux dimensions*

Soit T un tableau à deux dimensions de $n \times n$ cases, identifiées par $T[i, j]$ (pour $1 \leq i, j \leq n$).

1. Grâce à deux boucles imbriquées, écrire un algorithme renvoyant le maximum du tableau. Quelle est la complexité de cet algorithme ?
2. Afin de trouver le maximum, on peut également procéder ainsi : couper le tableau en quatre parties, trouver récursivement les maximums m_1, m_2, m_3, m_4 de chaque partie, puis renvoyer le maximum de m_1, m_2, m_3 et m_4 . Écrire un tel algorithme récursif.
3. En utilisant le théorème du coût récursif, évaluer la complexité de l'algorithme récursif précédent.

Exercice 3.*Tri rapide*

Le *tri rapide* est un algorithme de tri récursif d'un tableau T (à une dimension) dont le principe est le suivant. Le premier élément a du tableau est choisi comme *pivot* ; on réordonne ensuite les autres éléments de T pour placer tous les éléments $x \leq a$ avant a et les éléments $x > a$ après a . Ainsi, le tableau se décompose maintenant en T_1, a, T_2 , où T_1 contient uniquement des éléments inférieurs à a et T_2 uniquement des éléments supérieurs à a . Enfin, on appelle récursivement la procédure pour trier T_1 et T_2 .

1. Exécuter l'algorithme sur le tableau [10, 5, 3, 12, 8, 17].
2. Écrire l'algorithme de tri rapide.
3. Quelle est la complexité du tri rapide dans le pire cas ?
(Note : en moyenne, la complexité est $O(n \log n)$)