

TD3 : Récursivité**Exercice 1.**

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.

Exercice 2.

Appliquer le Master Theorem sur les cas suivants.

1. $T(n) = 9T(n/3) + n$
2. $T(n) = T(2n/3) + 1$
3. $T(n) = 3T(n/4) + n \log n$
4. $T(n) = 2T(n/2) + n \log n$
5. $T(n) = 2T(n/2) + n^3$
6. $T(n) = T(9n/10) + n$
7. $T(n) = 7T(n/3) + n^2$
8. $T(n) = T(\sqrt{n}) + 1$

Exercice 3.

Dans cet exercice, on s'intéresse au problème de l'élément majoritaire dans un tableau. On nous donne en entrée un tableau \mathbf{t} de n éléments. Le but est de déterminer s'il existe un élément qui apparaît strictement plus que $n/2$ fois dans \mathbf{t} , auquel cas on renvoie sa position, autrement on renvoie -1 .

1. Écrire un algorithme naïf `ElementMajoritaire` pour résoudre ce problème, qui calcule pour chaque position $i \in \{0, \dots, n-1\}$ le nombre d'occurrences de $\mathbf{t}[i]$ dans \mathbf{t} , et conclut en conséquence. Quelle est sa complexité ?
2. On cherche maintenant à obtenir un algorithme de meilleure complexité en utilisant la méthode Diviser pour Régner.
 - (a) Posons une position i . On considère le sous-tableau \mathbf{t}_0 qui contient les éléments $\mathbf{t}[0], \dots, \mathbf{t}[i-1]$ et le sous-tableau \mathbf{t}_1 qui contient les éléments $\mathbf{t}[i], \dots, \mathbf{t}[n-1]$. Prouver que si x est un élément majoritaire de \mathbf{t} , alors x est aussi un élément majoritaire d'au moins un des deux sous-tableaux \mathbf{t}_0 ou \mathbf{t}_1 .
 - (b) Écrire une version récursive de `ElementMajoritaire` qui repose sur le principe Diviser pour Régner, à l'aide de l'observation précédente.
 - (c) Utiliser le Master Theorem pour calculer la complexité de cette version récursive de `ElementMajoritaire`.