

ALGO3 – Algorithmique et Programmation 3

TP2 : Complexité & Récursivité**Récursivité**

1. Écrire une fonction récursive qui calcule la somme des n premiers termes d'une suite arithmétique de premier terme a et de raison r données.
2. Écrire une fonction récursive qui calcule le pgcd de deux entiers naturels a et b par la méthode d'Euclide.
3. Écrire une fonction récursive qui inverse une chaîne de caractères.
4. Écrire une fonction booléenne et récursive qui teste si une chaîne de caractères donnée est un anagramme d'une autre chaîne de caractères donnée. Par exemple : 'algorithm' est un anagramme de 'logarithme'.
5. Écrire une fonction récursive qui retourne la notation binaire d'un entier naturel. Par exemple : $25 = 1.2^4 + 1.2^3 + 0.2^2 + 0.2^1 + 1.2^0 \rightarrow 11001$.
6. Écrire une fonction récursive qui retourne la notation décimale d'un nombre binaire. Par exemple : $11001 \rightarrow 25$.
7. Écrire un programme récursif, puis non récursif pour savoir si une chaîne de caractères est un palindrome c'est à dire peut se lire dans les deux sens comme '121', 'abba'.
8. Soient deux listes d'entiers triées $liste_1$ et $liste_2$. Écrire la fonction `fusion` qui retourne une nouvelle liste triée contenant les éléments des listes $liste_1$ et $liste_2$.

Complexité

1. Étant donné une liste d'entiers dont chacun apparaît deux fois excepté un qui apparaît une seule fois. Écrire un programme permettant de trouver ce nombre.
2. Étant donné une liste d'entiers, trouver le premier élément qui se répète.
3. Étant donné une liste d'entiers non triés, écrire un programme permettant de tester si la liste contient tous les entiers d'un certain intervalle.
4. Étant donnée une liste A , écrire un programme permettant de trouver une instance de i, j, k tel que $0 \leq i < j < k < n$ et $A[i] < A[j] < A[k]$.