

Travaux Pratiques d'Algorithmique n° 2

Récurtivité

Le but de ce TP est de programmer, étudier et comparer des fonctions écrites de manière récursive et de manière itérative

1. Ecrire pour chacune des fonctions suivantes une version itérative et une version récursive. Comparez le nombre de calculs effectués dans chacun des cas. Dans le cas de la version récursive, calculez également le nombre d'appels récursifs effectués.
 - a. Fonction factorielle.
 - b. Fonction `sommeChiffre` qui retourne la somme des chiffres d'un nombre dans son écriture en base 10.
(Exemple : `sommeChiffre(124) = 7`)
 - c. Fonction qui calcule le $n^{\text{ième}}$ terme de la suite de Fibonacci définie récursivement par :

$$f(n) = f(n - 1) + f(n - 2) \quad \text{et} \quad f(0) = f(1) = 1.$$

2. Ecrire une fonction calculant les termes de la fonction d'Ackermann (voir poly) et affichant le nombre d'appels effectués.
3. Ecrire des versions récursives des tris par sélection et par insertion. Comparer en termes de rapidité et de nombre d'opérations de lecture et d'écriture aux versions itératives de la séance précédente.
4. Ecrire et comparer les versions itératives et récursives de la recherche d'un élément dans un tableau.
5. Ecrire et comparer les versions itératives et récursives pour l'inversion d'un tableau (le dernier élément devient le premier, l'avant-dernier devient le deuxième, etc...).