

TD 5 - La récursivité

Objectifs :

- Apprendre la récursivité (Condition d'arrêt de la récursivité + fonction récursive)
- Construire l'arbre des appels récursifs

1. Principe de la récursivité

Un sous-programme récursif est un sous-programme qui contient dans son code un appel à lui-même.

```
sous_prog FRECUR (.....)
  Var
  Début
    ...
    FRECUR (.....)
    ...
  Fin
```

Questions :

Comment s'arrête le sous-programme ?
Qu'est ce qui est récursif dans le problème ?

2. Exemple : La factorielle

$n! = n * (n-1)! = n * (n-1) * (n-2)! = \dots = n * (n-1) * (n-2) * \dots * 2 * 1 * 0!$

donc condition d'arrêt $n = 0$ et fonction récursive $f(n) = n * f(n-1)$

3. Enoncé

Effectuer les sous-programmes récursifs suivants.

3.1. Suite de Fibonacci

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2} \text{ pour } n > 1$$

3.2. Palindrome

3.3. PGCD

$$\text{pgcd}(a, 0) = a$$

$$\text{si } a > b \text{ alors } \text{pgcd}(a, b) = \text{pgcd}(a-b, b) \text{ sinon } \text{pgcd}(a, b) = \text{pgcd}(b-a, a)$$

3.4. Calcul de combinaison

$$C_{(n, 0)} = 1$$

$$C_{(0, p)} = 0$$

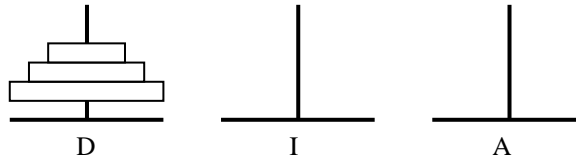
$$C_{(n, n)} = 1$$

$$C_{(n, p)} = C_{(n-1, p-1)} + C_{(n-1, p)} \quad \text{pour } n \text{ et } p \text{ donnés, avec } 0 < p < n$$

3.5. Ecrire avec un programme récursif x^n

3.6. Les tours de Hanoï

Trois tours D, I, A (Départ, Intermédiaire et Arrivée) et n rondelles empilées par taille décroissante sur la première tour. Il faut faire passer toutes les rondelles sur la dernière tour toujours empilées par ordre croissant, sachant qu'on ne peut bouger qu'une rondelle à la fois et qu'il faut conserver systématiquement un empilage décroissant des rondelles.



Faire un programme récursif qui indique tous les déplacements à faire.

3.7. Le tri rapide

Le tri à bulles est efficace pour des tableaux de petites tailles mais se révèle extrêmement lent pour des tableaux plus conséquents. En VB, à titre comparatif, trier un tableau de 50 000 entiers prends plus de 23 secondes avec un tri à bulles, plus de 16 secondes avec la version améliorée pour environ 10 à 25 millisecondes avec le tri rapide que nous allons voir ici (ce qui en fait l'algorithme de tri le plus utilisé actuellement).

L'idée du tri rapide est de séparer le tableau en deux sous-tableaux autour d'une valeur intermédiaire appelée pivot et de recommencer le processus récursivement. On choisit une valeur pour le pivot (la première valeur, la dernière ou une valeur au hasard dans le tableau) et on crée deux sous-tableaux : le premier contient toutes les valeurs inférieures au pivot et le second toutes les valeurs supérieures ou égales au pivot. Ainsi, on obtient deux nouveaux tableaux à trier mais de taille plus petite et l'on sait de plus que le pivot est correctement placé. Le processus peut se définir comme suit :

- Choix d'un pivot (le premier ou le dernier élément par exemple)
- Partitionnement en deux sous-tableaux (on échange les éléments du tableau de sorte que le début du tableau contienne tous les éléments inférieurs au pivot et que le pivot soit bien placé)
- Tri des deux sous-tableaux par la même méthode

Implémenter le tri rapide pour un tableau d'entiers.