L2 Programmation Impérative

TD2 Récursivité

1 Affichage

Afficher les caractères lus du clavier jusqu'au caractère `\n` sur une ligne dans l'ordre de lecture et sur la ligne suivante dans l'ordre inverse. Par exemple, si on a lu a b c d \n , l'affichage serait sur une ligne a b c d et sur la ligne suivante d c b a.

2 Conversion en base 2

Ecrire une fonction qui prend comme paramètre un entier positif n et qui affiche sur l'écran n en base 2.

3 Schéma de Horner

Ecrire une fonction récursive pour évaluer un polynôme avec le schéma de Horner.

4 Recherche récursive MAX

Nous nous intéressons à une solution récursive du problème de trouver la valeur maximale dans un tableau. A chaque appel, on divise le tableau en 2 sous-tableaux de taille égale (ou égale à un élément près). La fonction MAX2 prendra 3 paramètres : un tableau d'entiers, T ; un entier qui indique l'indice du premier élément du tableau, g ; un entier qui indique l'indice du dernier élément du tableau, d.

$$\begin{array}{lll} Max2(T,g,d) = & max\,\{Max2(T,g,\lfloor (g+d)/2 \rfloor), Max2(T,\lfloor (g+d)/2 \rfloor + 1, d)\} & g > d \\ Max2(T,g,d) = & T[g] & g = d \end{array}$$

Pour le premier appel $g = 0$ et $d = n - 1$ pour T de l'indice 0 à $n - 1$ de taille n.

1. Ecrire une fonction récursive Max2. Calculer le nombre de comparaisons des éléments du tableau si la taille est une puissance de 2 ($n = 2^k$).
2. Ecrire une fonction pour trouver d'une façon séquentielle la valeur maximale dans un tableau. Calculer le nombre de comparaisons des éléments du tableau pour un tableau de taille n.
3. Quelle solution préférez vous ?
4. On suppose que les éléments du tableau sont ordonnés du plus petit au plus grand et on cherche un élément X dans ce tableau : si $X \in T$ la fonction retourne l'indice de X ; sinon elle retourne -1. Comparez les deux solutions (recherche dichotomique, recherche séquentielle) pour ce cas.

5 Les tours de Hanoï

On dispose de 3 piquets numérotés 1,2,3 et n rondelles de tailles différentes entourant le piquet 1, avec la plus grosse en bas et la plus petite en haut. On veut amener les rondelles du piquet 1 au piquet 3 en ne prenant qu'une seule rondelle à la fois et en s'arrangeant pour qu'à tout moment il n'y ait jamais une rondelle sous une plus grosse. Un raisonnement par récurrence permet de trouver la solution :

– si $n = 1$ c'est trivial

- si $n > 1$:
  - Amener les $(n-1)$ disques du piquet *origine* vers le piquet *intermediaire*
  - Prendre la plus grande rondelle en bas de *origine* et la mettre tout seul en *destination*
  - Amener les $(n-1)$ rondelles de *intermediaire* en *destination*

1. Ecrire la fonction récursive et faire un appel dans le programme principal pour déplacer 8 rondelles du piquet 1 vers le piquet 3.
2. Essayer d'écrire le même programme sans utiliser la récursivité.