Travaux Pratiques d'algorithmique n°4

Cours d'Informatique de Deuxième Année

—DEUG Sciences M3-MI—

Récursivité

Dans cette séance, nous approfondissons la notion de récursivité à travers deux exemples : la recherche dichotomique, et le problème des tours de Hanoï. Nous nous intéressons en particulier aux calculs de complexité de chaque algorithme implémenté.

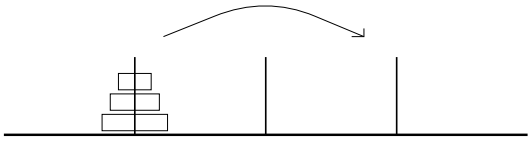
▸ Exercice 1. Recherche d'un élément dans un tableau trié

- *Écrire une fonction* `Recherche` *qui prend en argument un tableau* `tab` *de réels, sa taille* `n` *et un élément* `x` *et qui renvoie la position de* `x` *dans* `tab`*, ou* -1 *si* `x` *n'est pas dans le tableau. Evaluer la complexité de la fonction ;*
- *écrire une fonction récursive* `RechercheRec`*, qui effectue la même opération que précédemment, mais cette fois-ci en utilisant la méthode dichotomique. Evaluer la complexité ;*
- *écrire une fonction* `MaxMin` *qui implémente la recherche du maximum et du minimum du tableau. Que pensez-vous de cette fonction ?*

▸ Exercice 2.

Nous cherchons à résoudre le problème des "tours de Hanoï". Ce problème consiste à déplacer `n` *disques d'un piquet de départ à un piquet d'arrivée en utilisant un troisième piquet. La seule contrainte tient au fait qu'il est interdit de poser un disque sur un disque plus petit que lui.*

Résolution du problème pour $\mathtt{n} = 2$.



Exemple du problème pour $\mathtt{n} = 3$.

1. *résoudre "à la main" l'algorithme pour* `n` $= 3$ *;*
2. *en déduire un algorithme récursif de résolution du problème ;*
3. *écrire la fonction correspondant à cet algorithme : la fonction devra afficher la suite des déplacements des disques ; elle retournera le nombre de déplacements nécessaires.* `int Hanoi(int n, char dep, char tmp, char arr)` *où* `n` *est la hauteur de la tour, et* `dep,tmp,arr` *sont les noms (un seul caractère) des trois piquets de départ, de transition, et d'arrivée ;*
4. *donner enfin, la complexité en temps de votre algorithme.*