

# Cours d'Algorithmique

## Troisième fiche de TD — Les arbres, le back-track

Département Info ESIL — usage interne

2006-07

Une des principales opérations est le parcours d'un arbre. Parcourir un arbre, c'est visiter tous ses nœuds depuis la racine jusqu'aux feuilles, en empruntant les "liens" qui conduisent d'un nœud à ses nœuds fils. En général, l'objectif du parcours est d'appliquer un certain traitement  $T$  aux nœuds de l'arbre au fur et à mesure qu'on y passe. Il y a 2 façons de procéder qui sont illustrées par les deux premiers exercices.

### 1 Parcours en profondeur

On explore l'arbre branche par branche, de la racine aux feuilles, en commençant par la branche la plus à gauche. On procède comme suit. Si l'arbre est réduit à une feuille on ne fait rien. Sinon, on effectue successivement les opérations qui consistent à appliquer  $T$  à la racine, puis récursivement au premier fils et ensuite récursivement au second fils et ainsi de suite s'il y a plus de deux fils.

### 2 Parcours en largeur

On explore l'arbre niveau par niveau, en commençant par la racine (niveau 1), puis les nœuds du deuxième niveau, puis ceux du troisième niveau, etc. Ce traitement fait naturellement appel aux files.

### 3 D'autres exercices

Écrire, pour des arbres binaires, les fonctions suivantes.

1. Calculer le nombre de nœuds internes de l'arbre.
2. Calculer le nombre de feuilles de l'arbre.
3. Calculer la hauteur de l'arbre.
4. Imprimer l'arbre sous-forme infixe, préfixe et post-fixe.
5. Énumérer et imprimer tous les sous-arbres.

Un algorithme de back-track parcourt récursivement un arbre abstrait (qui modélise un problème) à la recherche des solutions. Typiquement, on s'intéresse à des problèmes où, à chaque étape, on a le choix entre plusieurs voies pour parvenir à la solution. Cette méthode utilise intensivement la récursivité pour mémoriser les points de choix et savoir y revenir ultérieurement. C'est exactement ce que faisait Ariane déroulant son fil pour ne pas se perdre dans le Labyrinthe.

Attention, il n'y a pas de miracle : les temps de calcul sont souvent exponentiels, mais on résout des problèmes compliqués à cause de la combinatoire sous-jacente.

## 4 Plateau de pions

Sur un plateau comportant 7 cases alignées et numérotées de 1 à 7 se trouvent, à gauche, 3 pions blancs et, à droite, 3 pions noirs alors que la case du milieu est vide. Le jeu consiste à amener les pions noirs à la place des blancs et vice-versa, les règles étant les suivantes.

On peut déplacer un pion de la case  $n$  à la case  $n + 1$  ou  $n - 1$  si la case d'arrivée est vide. On peut déplacer un pion de la case  $n$  à la case  $n + 2$  ou  $n - 2$  si la case d'arrivée est vide et si la case  $n + 1$  ou  $n - 1$  est occupée par un pion d'une autre couleur.

Écrire l'algorithme qui donne toutes les possibilités d'arriver à la configuration finale. Réfléchir à ce qui suit.

- Comment représenter le plateau ?
- Comment représenter les déplacements possibles ?
- À quelle condition un déplacement est-il acceptable ?
- Comment représenter (et conserver) la suite des déplacements qui aboutit à une solution ?

## 5 Problèmes des 8 dames

Comment placer 8 dames sur un échiquier sans qu'aucune ne soit en prise vis-à-vis d'une autre ? Il y a 92 solutions, si on ne tient pas compte des diverses symétries. Écrire l'algorithme qui les calcule.