TD d'algorithmique avancée
TD 8 : Dénombrement sur les arbres binaires

Jean-Michel Dischler et Frédéric Vivien

Dénombrement sur les arbres binaires

Dans cet exercice on notera n le nombre de nœuds d'un arbre binaire, f son nombre de feuilles et h sa hauteur. Tous les arbres considérés seront supposés non vides.

1. Quelle est la hauteur maximale d'un arbre à n nœuds ?
2. Quel est le nombre maximal de feuilles d'un arbre de hauteur h ?
3. Quel est le nombre maximal de nœuds d'un arbre de hauteur h ?
4. Quelle est la hauteur minimale d'un arbre de n nœuds ?
5. Montrez que le nombre de branches vides — nombre de fils gauches et de fils droits vides — d'un arbre à n nœuds est égal à $n+1$.

   Indication : on distinguera les nœuds ayant zéro, un et deux fils.
6. Montrez que le nombre de feuilles est inférieur ou égal à $\frac{n+1}{2}$ ($f \leq \frac{n+1}{2}$) et qu'il y a égalité si et seulement si chaque nœud de l'arbre est soit une feuille, soit a deux fils.

   Indication : se servir du raisonnement utilisé à la question précédente.
7. Montrez que le nombre de feuilles d'un arbre est égal au nombre de nœuds de degré deux, plus un.

Complexité du tri

Les algorithmes de tris par comparaison peuvent être considérés de façon abstraite en termes d'**arbres de décision**. Un arbre de décision représente les comparaisons (à l'exclusion de toute autre opération) effectuées par un algorithme de tri lorsqu'il traite une entrée de taille donnée. La figure 1 présente l'arbre de décision correspondant à l'algorithme de tri par insertion s'exécutant sur une liste de trois éléments.

Soient $\langle a_1, a_2, ..., a_n \rangle$ les n valeurs à trier. Dans un arbre de décision chaque nœud interne est étiqueté par une expression $a_i \leq a_j$, pour certaines valeurs de i et de j, $1 \leq i, j \leq n$. L'exécution de l'algorithme de tri suit un chemin qui part de la racine de l'arbre de décision pour aboutir à une feuille. À chaque nœud interne, on effectue une comparaison $a_i \leq a_j$ et les comparaisons suivantes auront lieu dans le sous-arbre gauche si $a_i \leq a_j$ et dans le sous-arbre droit sinon. Chaque feuille est désignée par une permutation $\langle \pi(1), \pi(2), ..., \pi(n) \rangle$: si l'algorithme de tri aboutit en la feuille $\langle \pi(1), \pi(2), ..., \pi(n) \rangle$, les valeurs à trier vérifient : $a_{\pi(1)} \leq a_{\pi(2)} \leq ... \leq a_{\pi(n)}$.

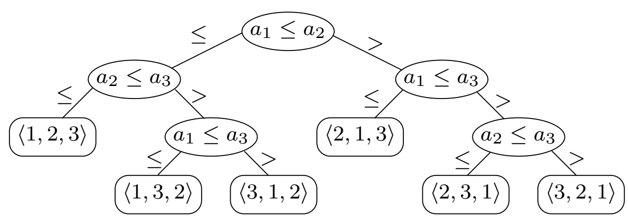

Fig. 1 – Arbre de décision correspondant au traitement de trois éléments au moyen du tri par insertion.

1. Quel est le nombre de feuilles d'un tel arbre de décisions ?
2. En déduire une borne inférieure sur la hauteur de l'arbre de décision.
3. En déduire une *borne inférieure* sur la complexité du tri de n éléments.
   Indication : d'après la formule de Stirling, on a $n! > \left(\frac{n}{e}\right)^n$.

Arbres binaires de recherche

1. Montrez que le temps de création d'un arbre binaire de recherche à partir d'une liste quelconque de n éléments est $\Omega(n \log n)$.
2. Écrivez un algorithme qui teste si un arbre binaire est un arbre binaire de recherche.