
INF233-TD8-PARCOURS D'ARBRES II

Tout ce qu'il faut savoir sur les parcours d'arbres

18 octobre 2011

Deux types de parcours : en *profondeur* et en *largeur*.

Parcours en profondeur : DFS

On parcourt le sommet le plus profond d'abord.

Schéma général :

- Pré-traitement de la racine
- Traitement récursif des enfants de la racine
- Post-traitement de la racine

Structure de données : On utilise une file (les fils sont traités de gauche à droite).

Besoin en espace : de l'ordre de la profondeur de l'arbre

Utilisation : Pour tous les problèmes qui se traitent bien de manière récursive : nombre de noeuds, profondeur max, nombre de feuilles, maximum des valeurs des noeuds,...

Variantes : Parcours préfixe, postfixe, infixe (infixe seulement pour les binaires).

Parcours en largeur : BFS

On parcourt les sommets niveau par niveau.

Schéma général :

- Deux couches : couche courante (CC) et couche des enfants (CE).
- Départ : $CC = [racine]$, $CE = vide$
- On traite les éléments de la CC. Pour chaque élément, on ajoute ses enfants à la CE.
- Quand la CC est vide on échange les deux couches
- On s'arrête quand les deux couches sont vides.

Structure de données : On utilise une file (premier entré dans une couche, premier traité).

Besoin en espace : de l'ordre de la largeur (taille maximum d'une couche) de l'arbre

Utilisation : Pour les problèmes qui ne se traitent pas bien de manière récursive, ou pour ceux où l'on a besoin d'information sur la profondeur des noeuds : lister les noeuds par profondeur croissante, donner la largeur de l'arbre...

Variantes : Une seule couche, traité comme une file (sinon on obtient un parcours en profondeur!), en ajoutant éventuellement des marqueurs pour séparer les couches.

Exercice 1 : Ecrire des algorithmes pour résoudre les problèmes suivants :

1. Trouver la feuille de profondeur minimale.
2. Calculer le nombre de feuilles de l'arbre.
3. Calculer la largeur de l'arbre (nombre maximum de noeud dans un niveau).
4. On suppose que chaque noeud contient une valeur. Calculer la valeur maximum des noeuds.
5. On suppose que chaque noeud contient une valeur. Calculer la deuxième valeur maximum des noeuds.
6. On suppose que chaque noeud contient une valeur. Donner la profondeur pour laquelle la somme des noeuds est maximum.
7. On suppose que l'on a un arbre binaire représentant une expression arithmétique utilisant les opérateurs suivants : $+$, $*$, $-$, $/$. Les feuilles contiennent des nombres et chaque noeud interne contient un opérateur. Ecrire un algorithme qui calcule la valeur de l'expression arithmétique.