Algorithmique
TP : Structure de données d'arbre

Raphaël Ginoulhac

September 21, 2019

1 Classe d'arbres

2 Affichage d'un arbre

La suite 12 8 4 9 23 17 15 correspond à un parcours en profondeur d'abord en entrant.

3 Gestion d'erreur

Les différents cas d'erreur :

- getSon : pos n'est pas entre 0 et nbSons-1, on veut alors accéder a un élément qui n'est pas dans le vecteur
- setSon : même chose
- removeLastSon : il n'y a aucun fils c'est-à-dire $nbSons \leq 0$

On implémente et documente le signalement d'erreur par exception et on laisse tout le code du test de rattrapage d'erreur à la fin du main.

4 Templatisation

On ne peut pas séparer $Tree < T >$ en 2 fichiers Tree.h et Tree.cpp pour la compilation séparée car il s'agit d'une classe Template. On peut garder la gestion d'erreur par exception.

5 Différents parcours d'arbres

On implémente les parcours en profondeur d'abord en entrant, en sortant, et le parcours en largeur. Les parcours en profondeur sont très simples car une boucle for suffit, et seule la position de l'affichage dans la fonction récursive

change. Le parcours en largeur nécessite de créer une file, et à chaque étape on prend le premier élément de la file, on l'affiche et on ajoute ses fils à la file. On initialise la file avec la racine de l'arbre.

La fonction maxDepth calcule rapidement et sans allouer de mémoire la profondeur maximale de l'arbre : on compare la profondeur maximale de chaque fils du noeud courant, et on lui ajoute 1. On utilise un parcours en profondeur d'abord car on va ainsi visiter les feuilles les plus profondes le plus rapidement.

La fonction minDepth calcule rapidement la profondeur minimale de l'arbre. On initialise deux files q_tree et q_count par la racine de l'arbre, et 1. Ces files vont contenir respectivement un sous-arbre de l'arbre initial et la profondeur de la racine de ce sous-arbre. Ensuite, à chaque étape, on prend le premier élément de chaque file, notés current et count. Si l'arbre current n'a pas de fils, alors on renvoie count, qui correspond à sa profondeur. Sinon, on ajoute ses fils à q_tree et autant de (count+1) à q_count.

On utilise un parcours en largeur car on peut détecter la feuille de profondeur minimale le plus rapidement avec ce parcours.