IMA 3ème année

Programmation Avancé

TP8 Arbres d'entiers

1 Objectifs

— Savoir créer et utiliser une bibliothèque d'arbres d'entiers
— Savoir coder quelques algorithmes de base sur les arbres

Préparation : Avant de commencer, copier le répertoire `~wrudamet/public/IMA3/TP8/` qui contient :
— Les fichiers `trees.c` et `trees.h`, contenant des fonctions de base de traitement d'arbres
— Un fichier `main.c` utilisant ces fonctions
— Un sous-répertoire `samples/` qui contient des fichiers de données `.txt` et des fichiers `.dot` et `.pdf` qui permettent de visualiser les arbres correspondant à ces données (section 2).
— Un sous répertoire `part2/` contenant un autre `main.c`.

2 Fonctions de base dans les ABR

Dans cette première partie, il vous est demandé d'implémenter les fonctions suivantes non implémentées dans `tree.h`, en utilisant le `main` fourni pour tester :

1. Ajout d'un élément dans un ABR (arbre binaire de recherche, donc ordonné !). (Fait en cours)
2. Construction d'un ABR à partir d'un fichier de données. Pour cette fonction, on s'inspirera de la fonction `charge_fichier` donnée dans le TP7 (attention, on ajoute des entiers et non des chaînes)
3. Impression d'un ABR par parcours récursif (fait en cours)
4. Vérifier que la fonction de parcours précédente donne la même suite de valeurs pour les trois fichiers `unspecified.txt`, `balanced.txt`, `degenerated.txt` alors que ce sont des arbres différents (voir les dessins dans le répertoire `samples/`)
5. Écrire une autre fonction d'impression qui imprime les couples (père, fils) d'un arbre donné. On fera attention à ne pas imprimer les sous-arbres vides
6. Vérifier que l'impression donne des arbres correspondant aux fichiers pdf fournis dans le répertoire `samples/`
7. Archiver le fichier `trees.o` dans une bibliothèque `libtrees.a` (commande `ar`).

3 Impression des arbres sous forme graphique

Les fichiers `.pdf` ont été générés à partir des fichiers `.dot` (fournis dans `samples/`). Le format `.dot` est un format de représentation textuelle simple d'arbres [1]. Visualisez le format de ces fichiers dans votre éditeur de texte préféré ou par la commande `less` [2]. À partir de ce format `.dot`, la commande `dot` permet de générer une version visualisable (.pdf, .png, ...) comme suit (en pdf par exemple) :

```
dot -Tpdf samples/balanced.dot -o samples/balanced.pdf
```

L'objectif est alors de générer de tels fichiers pour des arbres construits à partir de fichiers de données (tels que ceux fournis dans `samples/*.txt`).

Préparation : Vous pouvez copier votre bibliothèque `libtrees.a` dans le répertoire `part2/`. Dans ce répertoire, un `main.c` plus conséquent est fourni qui :

1. Génère, par appel à la fonction `generate_dot`, les fichiers `.dot` correspondants aux fichiers de données dont les noms ont été passés en paramètre.
2. Transforme ces fichiers `.dot` en `.pdf` par appel système à la commande `dot`.

Travail à faire :

1. Programmer la fonction `recursive_dot` qui traite le cas général
2. Tester ensuite par la commande : `./trees ../samples/*.txt` Vous devez retrouver les fichiers `.dot` et `.pdf` fournis.

1. Et plus généralement de graphes exploitables par des logiciels tels que `http://www.graphviz.org`
2. digraph signifie *directed graph*, les arbres en font partie du fait de leur orientation `pere->fils`