TP numéro 2
Arbres Binaires de Recherche

Programmation avancée, ENSIIE

Semestre 2, 2010–11

Vous pouvez réaliser ce TP au choix en OCaml ou en C.

Exercice 1 : Interface

1. Écrire le fichier d'interface des dictionnaires (fichier `dictionnaire.[mli|h]`). Il contient le type abstrait des dictionnaires, et les quatres prototypes des fonctions de manipulation des dictionnaires `creer`, `rechercher`, `inserer` et `supprimer`.
2. Écrire un fichier `test.[ml|c]` qui déroulera le scénario suivant :
   – créer un dictionnaire vide de taille 10 ;
   – ajouter l'association `3` $\mapsto$ `"coucou"` dans le dictionnaire ;
   – vérifier que la recherche de `3` dans le fichier renvoie bien `"coucou"` ;
   – ajouter l'association `1` $\mapsto$ `"toto"` dans le dictionnaire ;
   – ajouter l'association `2` $\mapsto$ `"titi"` dans le dictionnaire ;
   – vérifier que la recherche de `1` dans le fichier renvoie bien `"toto"` ;
   – supprimer `1` dans le dictionnaire ;
   – vérifier que la recherche de `3` dans le fichier renvoie bien `"coucou"`.
3. Écrire le `Makefile` correspondant. Indication : `test.cmo` (resp. `test.o`) dépend de `dictionnaire.mli` (resp. `dictionnaire.h`).

Exercice 2 : Arbres Binaires de Recherche

1. Dans un fichier `abr.[ml|c]`, implémenter les dictionnaires à l'aide d'arbres binaires de recherche.
2. Modifier le `Makefile` pour permettre la compilation du programme de test utilisant cette implémentation.
   Indication : en OCaml, il faut copier `dictionnaire.mli` en `abr.mli`, et utiliser `Abr` au lieu de `Dictionnaire` dans le fichier `test.ml`.
3. Compiler, tester.
4. Modifier le fichier `abr.[ml|c]` pour utiliser des arbres bien équilibrés suivant la méthode AVL (rotations pour rééquilibrer l'arbre).