

TP7 : Arbres Binaires de Recherche

Installation

1. Récupérez l'archive `TP_Arbre_Binaire_Recherche.tar.gz` à partir :
 - de ma page *Enseignements* à
`http://www-info.univ-lemans.fr/~jacob/enseignement.html`
dans la rubrique *Programmation C*
 - du serveur de l'IUP à
`/info/tmp/AnnexesTPL2SPI/TP_Arbre_Binaire_Recherche/TP_ABR.tar.gz`
2. Décompressez la et désarchivez la.
3. Comme dans les TP précédents, normalement tous les programmes et modules peuvent être compilés par `make -f Makefile all`

Cadre

le but de ce TP est de vous montrer l'efficacité des ABR sur la structure de liste que nous avons vue précédemment. Pour cela, nous allons rechercher des mots dans un dictionnaire et établir le coût de ces recherches. On se propose ainsi de faire plusieurs réalisations de ce dictionnaire et de comparer les coûts de recherches entre ces réalisations. Le dictionnaire que nous allons manipuler dans ce TP est composé simplement de chaînes de caractères sans blancs, nous allons donc seulement utiliser le type `string_t` vu précédemment.

Sujet du TP

Q. 1 Réalisation avec une liste

Tout d'abord réalisez le dictionnaire avec le type `liste_t` que nous avons vu dans les précédents TP.

- (a) Exécutez le programme `dico_liste` avec différentes tailles de dictionnaire que vous pourrez trouver dans le répertoire Fichiers.
- (b) Déduisez en l'ordre de l'algorithme de recherche dans un tel dictionnaire.

Nous avons vu en cours que ce type de structure était le moins efficace pour la recherche d'un élément. Les coûts que vous avez obtenus vous serviront donc de référence pour vos prochaines réalisations (il faudra donc faire mieux).

Q. 2 Réalisation avec un ABR

- (a) Transformez votre type arbre binaire `ab_t` du TP précédent en un arbre binaire de recherche `abr_t`. Pour cela complétez les fonctions des fichiers `abr.[ch]`, `abr_fonctions.[ch]` et `noeud.[ch]`
- (b) Testez votre ABR sur les 3 types habituels avec les programmes de test `test_abr_individus`, `test_abr_fractions` et `test_abr_mystrings`
- (c) Utilisez ensuite le programme `dico_abr` pour tester l'efficacité de votre type `abr_t` dans la création du dictionnaire, la recherche et la suppression de mots dans celui-ci.

Q. 3 Réalisation avec un AVL

- (a) Transformez votre type arbre binaire `abr_t` en un arbre binaire de recherche équilibré `avl_t`
- (b) Utilisez alors le programme `dico_avl` pour tester l'efficacité de votre AVL

Rapport

- une archive contenant toutes les réalisations de vos arbres
- Une comparaison commentée des différentes mesures que vous avez faites sur vos réalisations de ce dictionnaire : liste, ABR et AVL