

Alire

Ce fichier respecte la norme DokuWiki

Realise par PARMENTIER Laurent, ELHIMDI Yasmine

Groupe 2.

Compte rendu du TP 2.

Difficultes

La principale difficulté connue est la compréhension des algorithmes proposés dans le fichier abr_tp.c. Après cette étape, le codage en langage C a été plus facile et plus rapide.

Bogues

La fonction bordure affiche des segmentations faults dans quelques cas. On a pas réussi à detecter d'où viens le problème. Sinon, pas d'autres bogues percus jusqu'à présent.

Trace d'execution

La création d'un arbre sans utiliser les commandes du menu proposé nous évite d'insérer à chaque fois des éléments lors de test des fonctions qui ont été modifiées.

```
a[0] = arbre_vide();  
a[0] = insere(a[0], 15);  
a[0] = insere(a[0], 10);  
a[0] = insere(a[0], 20);  
a[0] = insere(a[0], 5);  
a[0] = insere(a[0], 12);
```

```
a[0] = insere(a[0], 17);  
a[0] = insere(a[0], 25);
```

Resultat de l'arbre créé : Remarque : L'affichage se fait à partir de la fonction d'affichage donné en fichier annexe abr_tp.c

```
yasmine@yasmine-HP-Pavilion-dv6-Notebook-PC:~ensicaen.git/media/s2/  
algo_avancee/tps/tp2$ ./abr  
Commande ? a  
0 # arbre 0  
    25  
  20  
    17  
15  
    12  
  10  
    5  
  
Commande ? q
```

Exemple d'execution des fonctions : `parcour_RGD(a[0])`, `parcour_GRD(a[0])`, `parcour_GDR(a[0])` en utilisant l'arbre créé précédement (`a[0]`) :

```
yasmine@yasmine-HP-Pavilion-dv6-Notebook-PC:~ensicaen.git/media/s2/  
algo_avancee/tps/tp2$ ./abr  
Commande ? g  
0 # arbre 0  
15  
10  
5  
12  
20  
17  
25  
  
Commande ? r  
0 # arbre 0  
5  
10  
12  
15  
17  
20  
25  
  
Commande ? d  
0 # arbre 0  
5  
12  
10  
17
```

25
20
15

La commande 'c' permet d'exécuter la fonction `compter_RGD_GDR(a[0])`. Cette fonction est une fonction récursive qui incrémente un entier des qu'il y a un noeud. L'affichage préfixe et suffixe se fait en même temps.

```
yasmine@yasmine-HP-Pavilion-dv6-Notebook-PC:~ensicaen.git/media/s2/
algo_avancee/tps/tp2$ ./abr
Commande ? c
0 # arbre 0
prefixe: 15
prefixe: 10
prefixe: 5
suffixe: 5
prefixe: 12
suffixe: 12
suffixe: 10
prefixe: 20
prefixe: 17
suffixe: 17
prefixe: 25
suffixe: 25
suffixe: 20
suffixe: 15
nbr: 7
```

La commande 'p' affiche le résultat donné par la fonction `parcour_largeur(a[0])`. Pour cette fonction, il a suffi d'implémenter le code en C à partir de l'algorithme proposé, avec l'utilisation des fonctions nécessaires pour la manipulation des piles (CF TP1) :

```
yasmine@yasmine-HP-Pavilion-dv6-Notebook-PC:~ensicaen.git/media/s2/
algo_avancee/tps/tp2$ ./abr
Commande ? p
0 # arbre 0
15
10
20
5
12
17
25
```

A faire

valgrind

From:

<https://ensicaen.dokuwiki.society-lbl.com/> - **Ensicaen**

Permanent link:

https://ensicaen.dokuwiki.society-lbl.com/s2:algo_avancee:tp2:start

Last update: **2014/02/24 07:41**

