Mathieu Yanis

Martinez Baptiste

RAPPORT

Implémentations des fonctions :

-construit_arbre :

La fonction lit l'entrée à l'aide de fgets et la stocke dans le tableau « saisie ». Puis si le premier caractère de l'entrée est '0', elle définit le pointeur vers l'arbre « a » à NULL. Sinon, elle récupère l'étiquette du nœud à l'aide de « strtok », l'alloue avec « alloue_noeud », puis construit récursivement les sous-arbres gauche et droit en appelant récursivement « construit_arbre » pour avoir le parcours en profondeur préfixe. Et elle renvoie 1 si la construction de l'arbre est réussie, sinon 0 en cas d'échec.

-Serialise :

Pour la fonction « serialise » nous avons décidé de faire une fonction auxiliaire « serialise_aux ». La fonction « serialise_aux » prend un pointeur vers un fichier, un pointeur vers un nœud et un entier représentant l'indentation et renvoi un int. Si le nœud est NULL, elle écrit « NULL » dans le fichier et retourne 1. Sinon, elle écrit la valeur du nœud dans le fichier avec l'indentation appropriée, puis sérialise récursivement le sous-arbre gauche et le sous-arbre droit en incrémentant l'indentation. Elle retourne 1 si tout s'est bien passé, sinon 0.

La fonction « serialise » ouvre le fichier en mode écriture, vérifie si l'ouverture a réussi, puis appelle « serialise_aux » avec le fichier ouvert et l'arbre à sérialiser. Une fois la sérialisation terminée, elle ferme le fichier.

-deserialise :

Pour la fonction « deserialise » nous avons décidé de faire une fonction auxiliaire « deserialise_aux ». La fonction « deserialise_aux » prend un pointeur vers un fichier et renvoie un Arbre. Elle lit une ligne du fichier « f » dans le tableau ligne. Elle recupere l'étiquette du nœud en recherchant le premier

caractère « : » dans la ligne à l'aide de « strchr », puis avance de deux caractères pour obtenir l'étiquette. Elle ajuste ensuite la longueur de l'étiquette en remplaçant le caractère de nouvelle ligne par « \0 ». Puis elle alloue un nouveau nœud avec l'étiquette qu'on vient de récupérer. Elle lit la ligne suivante du fichier pour déterminer si le sous-arbre gauche est NULL ou non. Si ce n'est pas le cas, elle désérialise récursivement le sous-arbre gauche avec un appel à « deserialise_aux » et on fait de même pour le sous-arbre droit et on renvoie l'Arbre nouvellement créer.

La fonction « deserialise » ouvre le fichier en mode lecture, vérifie si l'ouverture a réussi, puis appelle « deserialise_aux » avec le fichier ouvert. Une fois la désérialisation terminée, elle ferme le fichier et elle vérifie si l'arbre créer est bien diffèrent de « NULL » si c'est le cas on renvoie 1 sinon 0.

-expansion :

Pour la fonction « expansion » nous avons décidé de faire une fonction auxiliaire « greffe ». La fonction «greffe» prend un pointeur vers un arbre G et deux autres arbres B et C : l'arbre G sur lequel on va effectuer la greffe, le sous arbre gauche B et le sous arbre droit C de l'arbre qu'on doit greffer. Elle parcours l'arbre G avec un parcours préfixe, c'est-à-dire qu'elle vérifie si la racine est nulle, si les fils gauche et droit sont nuls, ou si le fils gauche est nul ou si le fils droit est nul et ensuite si aucun n'est nul elle fait un appel récursif sur le fils gauche de G suivi d'un appel récursif sur le fils droit de G. Selon ce qui est nul, la fonction greffe va greffer sur le fils gauche de G, ou le fils droit, ou les deux le sous arbre gauche B, le sous arbre droit C, ou les deux à l'aide de la fonction copie.

La fonction « expansion » effectue un parcours suffixe sur l'arbre A donné en argument, c'est-à-dire qu'elle fait un appel récursif sur le sous arbre gauche de A jusqu'à ce qu'il soit nul puis un appel récursif sur le sous arbre droit de A jusqu'à ce qu'il soit nul. Ensuite elle vérifie si la valeur de A est égal à la valeur de B et si c'est le cas on procède à la greffe : elle fait une copie de l'arbre B qui est le greffon et appelle la fonction greffe avec la copie du greffon, le sous arbre gauche et le sous arbre droit de A pour faire une greffe sur le greffon. Puis pour finir la mémoire allouée pour A est libérée et A est égal à la copie du greffon.

Difficulté rencontrée :

On a rencontré certaines difficultés avec la fonction « construit_arbre ». Nous n'avons pas réussi à lire plusieurs chiffres simultanément lors de la saisie, c'est-à-dire que lorsque nous construisons un arbre, au lieu de saisir 0 0 1 « étiquette », nous devons saisir 0, puis 0, puis 1 « étiquette ».

La fonction expansion nous à également posé beaucoup de problèmes, si bien qu'un de nous deux a du s'attarder uniquement sur la fonction expansion et ce qui lui était relié pendant que l'autre avançait sur la suite du projet. Au final après plusieurs jours la fonction expansion fonctionne correctement et permet de faire de belles greffes. Ce qui a posé problème dans cette fonction est notamment la manipulation de pointeurs qui a posé de nombreuses erreurs de segmentation et trouver comment parcourir l'arbre dans la fonction expansion et dans la fonction greffe.

Répartition du travail :

A cause des problèmes causés par la fonction expansion et car il nous restait peu de temps, nous avons du faire en sorte que l'un de nous s'occupe uniquement de la fonction expansion pendant que l'autre s'occupait de la suite.