Algorithmique - TD4
Tas binaire

1 Introduction

Dans ce TP, vous allez implémenter en langage C un tas binaire représenté dans un tableau dont la taille est dynamiquement adpatée.

L'indice du fils gauche de la cellule i est $2i + 1$, celui du fils droit est $2i + 2$. Inversement pour trouver l'indice de la cellule parente de la cellule i, il suffit de calculer $\lfloor (i - 1)/2 \rfloor$. Un tas binaire représenté dans un tableau n'a pas de case inutilisée car les opérations que l'on effectue garantissent que l'on ajoute une case en fin de tableau (insertion) ou que l'on en supprime une en fin de tableau (extraction).

Les opérations de base que l'on peut effectuer sur le tas binaire sont les suivantes :

— Insertion d'une valeur. La valeur est d'abord positionnée à la fin du tableau puis des permutations sont effectuées tant que la propriété de tas binaire n'est pas satisfaite.

— Extraction du max. Cette opération renvoie la valeur maximum du tas (la racine) puis reconstruit le tas binaire en insérant la dernière valeur du tableau à la place de la racine et en faisant des permutations successives jusqu'à une feuille de l'arbre tant que la propriété n'est pas respectée.

A ces deux opérations de base, nous ajoutons deux opérations de gestion qui permettent d'allouer une taille initiale au tableau ainsi que la structure elle même, et une opération de destruction du tas binaire.

2 Travail à effectuer

La structure de données ainsi que l'ossature du programme vous sont données sur moodle. Vous devez les respectez. Voici la structure :

```
typedef struct {
  int allocated;
  int filled;
  int *array;
} BinaryHeap;
```

Le membre `allocated` indique le nombre de cases allouées dans le tableau `array`. Le membre `array` est un pointeur vers le tableau qui reçoit les valeurs du tas binaire. Par construction, la première valeur du tableau correspond à la racine donc à l'élément maximum du tas binaire. Le membre `filled` indique le nombre de cases effectivement remplies dans le tas binaire. L'algorithme d'insertion a été vu en cours, celui d'extraction n'a pas été donné de manière explicite. Vous devez bien réfléchir à cet algorithme, car il est moins trivial que l'insertion.

3 Description des entrées/sorties

Bien que ces entrées et sorties soient déjà données dans l'ossature de base du programme, en voici la description :

```
insert <n>
extract
bye
```

La commande `insert` permet d'ajouter une valeur au tas binaire. Elle prend en argument un entier qui est la valeur à ajouter. La commande `extract` permet de renvoyer la valeur maximum et de la supprimer du tas. Elle affiche cette valeur. La ligne affichée sera terminée par deux caractères (CR et LF), correspondant à la chaîne `"\r\n"`.

La commande `bye` permet de quitter le programme.

Un exemple d'entrée :

```
insert 15
insert 4
insert 8
insert 9
insert 2
insert 18
extract
extract
extract
extract
extract
extract
extract
bye
```

avec la sortie correspondante :

```
18
15
9
8
4
2
```

Dans cet exemple, la dernière commande `extract` n'aura aucun effet car il n'y a plus de valeurs dans le tas binaire.

Eric Guérin
eric.guerin@liris.cnrs.fr