

TDLOG – séance n° 6

Test : TP

Xavier Clerc – `xavier.clerc@enseignants.enpc.fr`

Cédric Doucet – `cedric.doucet@inria.fr`

Thierry Martinez – `thierry.martinez@inria.fr`

25 novembre 2015

À rendre au plus tard le 29 novembre 2015

L'objectif de cette séance de TP est d'écrire les classes et/ou fonctions *Python* permettant d'implémenter une file de priorité par le biais d'un tas binaire. **L'accent doit être mis sur les tests écrits afin de s'assurer de la correction du code.**

1 Tas binaire

1.1 Présentation

Un tas binaire est une structure de données arborescente qui vérifie les propriétés suivantes :

- il s'agit d'un arbre binaire parfait : tous les niveaux à l'exception du dernier sont complets, et sur le dernier niveau tous les nœuds sont à gauche ;
- la valeur de chaque nœud doit être supérieure ou égale à la valeur de chacun de ses fils.

La figure 1 montre un tas représenté comme un arbre.

Il est également possible de stocker l'arbre sous la forme d'un tableau, en utilisant les équations suivantes. Pour un nœud placé à l'index idx :

- $indice_parent = floor(idx/2)$;
- $indice_fils_gauche = 2 * idx + 1$;
- $indice_fils_droit = 2 * idx + 2$;

La figure 2 montre le même tas que précédemment représenté comme un tableau.

Dans le cadre de ce TP, on s'intéressera uniquement à deux opérations sur le tas :

- accès à l'élément de plus grande valeur (par définition la racine, placée à l'indice 0) ;
- ajout d'un élément au tas.

L'algorithme d'ajout est le suivant :

FIGURE 1 – Exemple de tas binaire comme arbre.

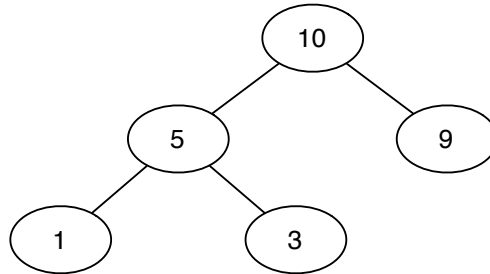


FIGURE 2 – Exemple de tas binaire comme tableau.

10	5	9	1	3
0	1	2	3	4

```
insertion(tableau, element):  
    ajouter element à la fin de tableau  
    idx = indice de l'élément ajouté  
    tant que (idx n'est pas la racine) et (valeur(idx) > valeur(parent(idx))) faire  
        échanger les valeurs aux indices idx et parent(idx)  
        idx = parent(idx)
```

2 File de priorité

Le principe d'une file de priorité est de maintenir une collection d'éléments *partiellement* ordonnés. On est typiquement intéressé par le fait d'accéder à l'élément de plus grande valeur.

Dans le cadre de ce TP, on souhaite programmer une file de priorité par le biais d'un tas binaire afin de déterminer quelle est l'information la plus *urgente* contenue dans un fichier.

Les fichiers qui nous intéressent sont de la forme suivante :

```
3: une information  
10: une autre information  
...  
4: une dernière information
```

soit un entier représentant l'urgence de l'information (plus l'entier est élevé plus l'information est importante) puis le caractère “:”, et enfin un message sous la forme d'une chaîne de caractères.

L'objectif est donc de programmer les éléments permettant de stocker les informations lues dans un fichier dans un tas binaire puis d'afficher l'information la plus urgente.

L'implémentation étant relativement simple, l'essentiel du travail porte en réalité sur les tests à mettre en place afin de s'assurer que les éléments développés sont corrects.