

Travaux Pratiques 6

Rappels de cours *Un tableau A est une suite finie de longueur(A) éléments dont le i ème élément est noté $A[i]$. Tout tableau A définit une structure d'arbre de racine $A[1]$ par la fonction père $p(i) = \lfloor i/2 \rfloor$ définie pour tout indice i de 2 à longueur(A). Cet arbre est binaire car chaque élément non-feuille a alors un fils gauche $g(i) = 2i$ ou/et un fils droit $d(i) = 2i + 1$. Si l'ensemble des éléments de A est totalement ordonné et si $A[p(i)] \geq A[i]$, alors A est un tas. On peut réordonner tout tableau de façon à en faire un tas en appliquant l'algorithme 1.*

algorithme 1 CONSTRUIRE-TAS(A)

```
taille[A]  $\leftarrow$  longueur[A]
for  $i \leftarrow \lfloor \text{longueur}[A]/2 \rfloor$  à (décroître) 1 do
    ENTASSER( $A, i$ )
end for
```

algorithme 2 ENTASSER(A, i)

```
 $l \leftarrow 2i$ 
 $r \leftarrow 2i + 1$ 
if  $l < \text{taille}[A]$  et  $A[l] > A[i]$  then
     $max \leftarrow l$ 
else
     $max \leftarrow i$ 
end if
if  $r \leq \text{taille}[A]$  et  $A[r] > A[max]$  then
     $max \leftarrow r$ 
end if
if  $max \neq i$  then
    échanger  $A[i]$  et  $A[max]$ 
    ENTASSER( $A, max$ )
end if
```

Exercice 1 Construction d'un tas

Question 1.1 Définissez la classe `Tas`, puis écrivez les fonctions suivantes :

- `def filsG(i)` qui, étant donné un indice i dans le tableau du tas, renvoie l'indice du fils gauche du noeud d'indice i .
- `def filsD(i)` qui effectue la même opération mais avec le fils droit.

Question 1.2 L'algorithme 2 permet de transformer une arborescence binaire dont la racine est l'élément d'indice i dans A en un tas. Écrivez une fonction `entasser(t,i)` qui réalise l'algorithme 2.

Question 1.3 L'algorithme 1 permet de construire un tas à partir d'un tableau d'entiers. Écrivez une fonction `construireTas(t)` qui réalise l'algorithme 1.

Exercice 2 Tri par tas

L'algorithme 3 est l'algorithme de **tri par tas** (*heapsort*) qui permet de trier un tableau d'entiers dans l'ordre croissant.

algorithme 3 TRIER-TAS(A)

```
CONSTRUIRE-TAS( $A$ )  
for  $i \leftarrow longueur[A]$  à 2 do  
    échanger  $A[1]$  et  $A[i]$   
     $taille[A] \leftarrow taille[A] - 1$   
    ENTASSER( $A,1$ )  
end for
```

Définissez et testez la fonction `def triParTas(t)` qui implémente l'algorithme de tri par tas.