

Entraînement : Arbres en structure de tas

Pour tous les exercices, la grille d'évaluation est la suivante.

Structure de données tas.

A (20)	Exemple tas / tableau correct
D (8)	Structure comprise mais erreurs (d'inattention ?)
E (1)	Exemples faux

Algorithme.

A (20)	L'algorithme répond correctement au problème posé, il est écrit de façon claire et la complexité est correcte.
B (16)	L'algorithme contient quelques erreurs mais reste globalement juste et la complexité est correcte.
C (11)	L'algorithme fonctionne globalement mais complexité non optimale ou complexité calculée fausse.
D (8)	L'algorithme ne fonctionne pas.
E (1)	Algorithme quasi inexistant ou ne répondant pas du tout au problème posé.

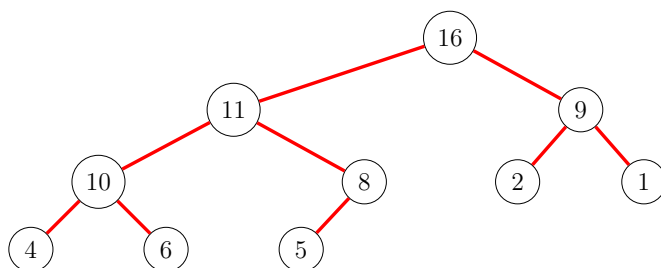
Exercice 1.

On rappelle qu'un tas est un **arbre binaire parfait décroissant** :

- tous les niveaux sauf le dernier sont remplis au maximum et les feuilles du dernier niveaux sont alignées à gauche,
- la valeur d'un nœud est toujours supérieure égale aux valeurs de ses fils.

La structure d'arbre binaire parfait permet au tas d'être stocké dans un tableau.

(1) Donner le tableau correspondant au tas suivant



(2) Dessiner l'arbre correspondant au tableau suivant

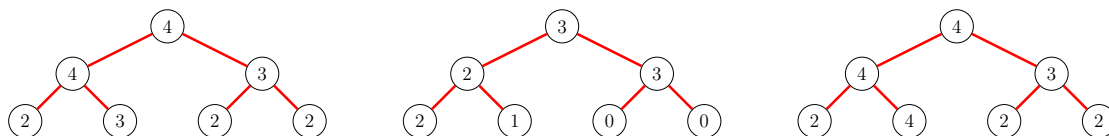
<i>indice</i>	0	1	2	3	4	5	6	7	8
<i>valeur</i>	5	4	2	3	2	1	1	2	1

On se place dans le contexte suivant : on possède un ensemble de serveurs qui gèrent des processus. Chaque serveur possède un certain nombre de *cœurs* libres pour traiter de nouveaux processus. On stocke la disponibilité des serveurs dans un tableau sous forme de tas, par exemple

0	1	2	3
4	2	2	0

signifie que j'ai 4 serveurs. Le serveur 0 possède 4 cœurs disponibles, les serveurs 1 et 2 en possèdent chacun 2 et le serveur 3, 0. Le serveur 0 sera toujours celui avec le plus de cœurs disponibles.

(3) Considérer les 3 exemples suivants :



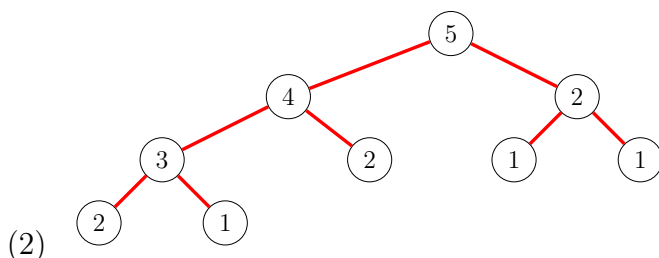
Pour chacun des exemples,

- Diminuer la valeur de $T[0]$ de 1 (on lui affecte un processus) et regarder si le résultat obtenu est un tas.
 - Si ce n'est pas le cas, décrire le processus à suivre pour *corriger* le résultat et obtenir un tas.
- (4) Implanter une fonction qui réalise de façon générale l'algorithme décrit sur les exemples précédents. Quelle est sa complexité ?

Solution

(1)

0	1	2	3	4	5	6	7	8	9
16	11	9	10	8	2	1	4	6	5



- (3) Aucun des trois ne reste un tas : il faut échanger la valeur avec son fils maximal puis recommencer récursivement sur le fils échangé.
- (4) Une solution possible

```

Input : Tableau T
Procédé :
  i ← 0
  Tant que 2*i+1 < T.taille :
    filsMax ← 2*i+1
    Si 2*i+2 < T.taille et T[2*i+2] > T[2*i+1] :
      filsMax ← 2*i+2
    Si T[i] < T[filsMax] :
      Echanger T[i], T[filsMax]
      i ← filsMax

```