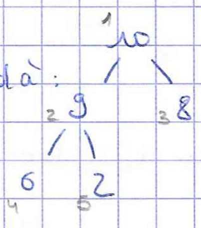
Correction et complexité du tri par tas

[Cormen p142-148] [Froidevaux ?]

def Un tas est 1 arbre binaire complet sauf éventuellement au dernier étage qui voit ses feuilles être le + à gauche possible

on le représente par 1 tableau A et 1 entier t = taille tq les élé du tas sont ds A[1..t] et tq, par ex :

$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ \end{array}$
[10, 9, 8, 6, 2] correspond à :



Au niveau des indices :

Parent(i) = $\lfloor \frac{i}{2} \rfloor$

Fils_Gauche(i) = $2i$

Fils_Droit(i) = $2i+1$

Un tas max est 1 tas tq la racine de tout sous arbre du tas porte 1 étiquette supérieure ou égale à celles des noeuds du sous arbre

(*) →

algo Entasser

Entrée = 1 tas A et un entier i, indice de A tq les fils gauche et droit du sous arbre enraciné en i (de racine l'élé d'indice i) sont des tas max

Sortie : une permutation de A tq l'arbre enraciné en i est 1 tas max (et dont les élé sont ceux initialement ds cet arbre)

```
Entasser (A, i) =
    g = Gauche(i) ; d = Droite(i) ; max = i
    Si g ≤ taille(A) et A[g] > A[i] alors
    |  max = g
    Si d ≤ taille(A) et A[d] > A[max] alors
    |  max = d
    Si max ≠ i
    |  échanger A[i] et A[max]
    |  Entasser (A, max)
```

} ainsi, max contient l'indice du + gd élé entre A[i], A[g] et A[d]

(sinon, A[i] est déjà plus gd que ses fils donc terminé)

Lemme La hauteur h d'un tas comptant n noeuds vérifie

$$\log_2(n+1)-1 \leq h \overset{<}{\leq} \log_2(n+1)$$

démo Soit T un tas de n éléments et h sa hauteur ( est de hauteur 1)

Ce tas est un arbre complet de hauteur $h-1$ auquel on a rajouté des feuilles au dernier étage. Donc nb feuilles arbre complet de taille $h-1$ $\overset{< \text{ en fait}}{\leq}$ n $\leq$ nb feuilles arbre complet de taille h

cad $\sum_{i=0}^{h-1} 2^i \overset{<}{\leq} n \leq \sum_{i=0}^{h} 2^i$

cad $2^h - 1 \overset{<}{\leq} n \leq 2^{h+1} - 1$ cad $h \overset{<}{\leq} \log_2(n+1) \leq h+1$

- Correction de l'algo OK : on fait descendre $A[i]$ dans l'arbre jusqu'à sa place
- Complexité pour un noeud i de hauteur h = $O(h)$

algo Construire_tas

Entrée : un tableau A

Sortie : une permutation de A qui en fait un tas max

Construire_tas(A) =

taille(A) = |A|

pour $i = \left\lfloor \frac{|A|}{2} \right\rfloor$ à 1 faire

| Entasser(A, i)

- Correction : on montre l'invariant suivant : "au début de la boucle pour, les noeuds $i+1, i+2, \dots n$ sont racines d'1 tas max"

→ Initialisation : on a $i = \left\lfloor \frac{|A|}{2} \right\rfloor$. Chaque noeud $i+1, \dots n$ est donc une feuille ; or une feuille est bien un tas max (car les fils des noeuds d'indices $\left\lceil \frac{|A|}{2} \right\rceil \dots$ auraient 1 indice > taille du tableau)

→ Hérédité : on passe de i à $i-1$. Les noeuds $i+1, \dots n$ sont racines de tas max à l'étape d'avant donc on est ds les hp pour appliquer Entasser(A, i) car les fils de i sont des tas max. Donc i est la racine d'1 tas max par correction de Entasser, ce qui assure par def tas max que les noeuds $i, \dots n$ sont racines de tas max

→ Fin : $i = 0$. En particulier 1 est racine d'1 tas max dc l'algo est correct.

- Complexité : on note $n = |A|$ (brutalement : $n \ln(n)$ ← cpxt Entasser)

Il y a au + et /rec $\left\lceil \frac{n}{2^{h+1}} \right\rceil$ noeuds de hauteur h et pour un noeud de hauteur h,

Entasser a 1 complexité en $O(h)$ et la hauteur d'1 noeud de A est

comprise entre 0 et $\lceil \log_2(n+1) \rceil$ par le lemme.

D'où 1 complexité en $\displaystyle\sum_{h=0}^{\lceil \log_2(n+1) \rceil} \left\lceil \frac{n}{2^{h+1}} \right\rceil O(h)$

$$= O\Big(n \underbrace{\sum_{h=0}^{\lceil \log_2(n+1) \rceil} \frac{h}{2^h}}_{\leq \sum_{h=0}^{\infty} \frac{h}{2^h} = 2 \text{ (dérivée série géométrique)}}\Big)$$

$$= O(n)$$

algo Tri par tas

Entrée : un tableau A

Sortie : le un tableau trié associé à A

```
Tri(A) =
    Construire_tas(A)
    pour i = |A| à 2 faire
        échanger A[1] et A[i]          → mettre le + gd au bout
        taille(A) = taille(A) - 1      → ne considérer que ce qui reste à trier
        Entasser(A, 1)                 → entasser pour garder 1 tas
```

- Correction OK

- Complexité $= O(n) + n \times O(\log_2(n))$

pour construire_tas ; boucle pour ; car Entasser(A, 1) se fait en O(hauteur A) et hauteur A $= \log_2(n)$

Donc complexité en $O(n \ln(n))$ et en $\Omega(n \ln(n))$ par th sur les tris par comparaison donc en $\Theta(n \ln(n))$

⚠ Il y a deux déf de hauteur ici !