

# GLIN 401 IE : Analyse algorithme

## 1 Notation O et $\Theta$

On rappelle les définitions de base : soit deux fonctions  $f(n)$  et  $g(n)$  de  $\mathbb{N} \rightarrow \mathbb{N}$ ,

$$f \in \Theta(g) \iff \{f \in O(g) \text{ et } g \in O(f)\} \text{ et}$$

$$f \in O(g) \iff \{\exists N_0 \in \mathbb{N}, \exists k \in \mathbb{N} \mid \forall n \geq N_0 : f(n) \leq kg(n)\}$$

**Question 1** justifiez vos réponses (vrai ou faux) aux deux questions suivantes :

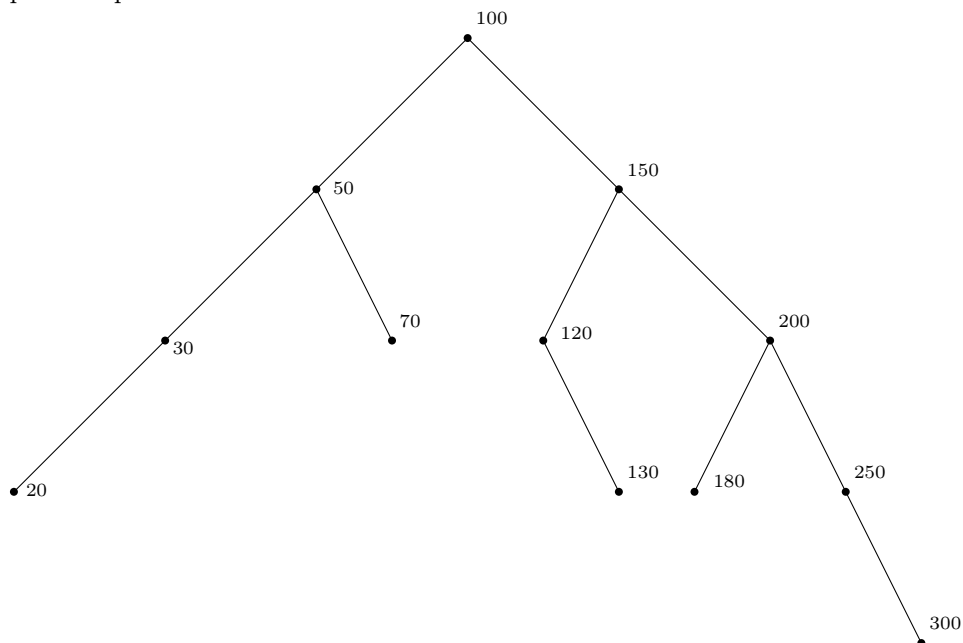
1. pour une constante  $k$  quelconque,  $k^{n+1} \in O(k^n)$
2. pour une constante  $k$  quelconque,  $n^k \in \Theta(n^{k+1})$

## 2 AVL

Il existe deux variantes dans l'algorithme qui supprime un sommet dans un ABR. Lesquelles ?

Choisir l'algorithme qui vous impose le moins de calcul pour supprimer le sommet de clé 50 dans l'AVL ci dessous. Justifier ce choix.

Indiquer alors chacune des étapes, d'abord pour supprimer le dit sommet, puis pour rééquilibrer l'AVL.



**Question 2** Dans l'arbre binaire de recherche que nous considérons dans ce problème, toutes les clés sont différentes.

Pour un sommet  $S$ , on notera  $cle(S)$  la clé de ce sommet.

1. à quelle condition un sommet  $S$  est-il le sommet ayant la plus grande clé ?  
Si  $S$  n'est pas le sommet qui a la plus grande clé, et si toutes les clés sont différentes, on appelle  $Succ(S)$  le sommet qui a la clé immédiatement supérieure à celle de  $S$
2. Dans le premier cas que nous examinons,  $S$  a une sous arborescence droite
  - (a) où pensez vous que se trouve alors  $Succ(S)$  ? appelez ce sommet  $Min(S)$  dans les explications suivantes par lesquelles vous allez démontrer que votre intuition est juste.
  - (b) expliquez pourquoi pour tout sommet  $T$ 
    - i. qui se trouve dans la sous arborescence droite de  $S$ ,  
 $T \neq Min(S) \Rightarrow T$  ne peut pas être le successeur de  $S$
    - ii. qui se trouve dans la sous arborescence gauche de  $S$ ,  $T$  ne peut pas être le successeur de  $S$
    - iii. qui se trouve sur le chemin de  $S$  à la racine,  $T$  ne peut pas être le successeur de  $S$  (il faudra examiner deux cas, suivant que  $S$  est dans la sous arborescence droite ou gauche de  $T$ )
    - iv. qui se trouve ailleurs dans l'arbre,  $T$  ne peut pas être le successeur de  $S$  (il faudra considérer  $U$  l'ancêtre commun à  $S$  et  $T$  le plus bas dans l'arbre, puis examiner deux cas)
3. Dans le deuxième cas que nous examinons,  $S$  n'a pas de sous arborescence droite :
  - (a) indiquez alors où vous pensez que se trouve  $Succ(S)$  et appelez le  $Bas(S)$  dans les explications ultérieures.
  - (b) prouvez le de la même manière que pour le premier cas

**Question 3** Dans un arbre binaire de recherche dont toutes les clés sont différentes, pour lequel on dispose des fonctions (en  $\Theta(1)$ )

- $racine?(S)$  qui indique si un sommet  $S$  est la racine de l'arbre
  - $pere(S)$ , définie pour les autres sommets que la racine et qui renvoie le père du sommet  $S$
  - $val(S)$  qui renvoie l'étiquette du sommet  $S$
  - $sad(S)$  qui renvoie la racine de la sous arborescence droite du sommet  $S$
  - $sag(S)$  qui renvoie la racine de la sous arborescence gauche du sommet  $S$
  - $fg?(S)$ , définie pour les autres sommets que la racine et qui indique si un sommet  $S$  est le fils gauche de son père
  - $fd?(S)$ , définie pour les autres sommets que la racine et qui indique si un sommet  $S$  est le fils droit de son père
  - $nonVide?(A)$  qui indique que l'arborescence  $A$  n'est pas vide
1. écrivez la fonction  $Succ(S)$  (une solution avec des fonctions auxiliaires est préférable).
  2. Donnez et justifiez la classe de complexité de votre fonction. (on choisira le bon paramètre, pour que votre fonction soit intéressante si l'ABR est aussi un AVL).