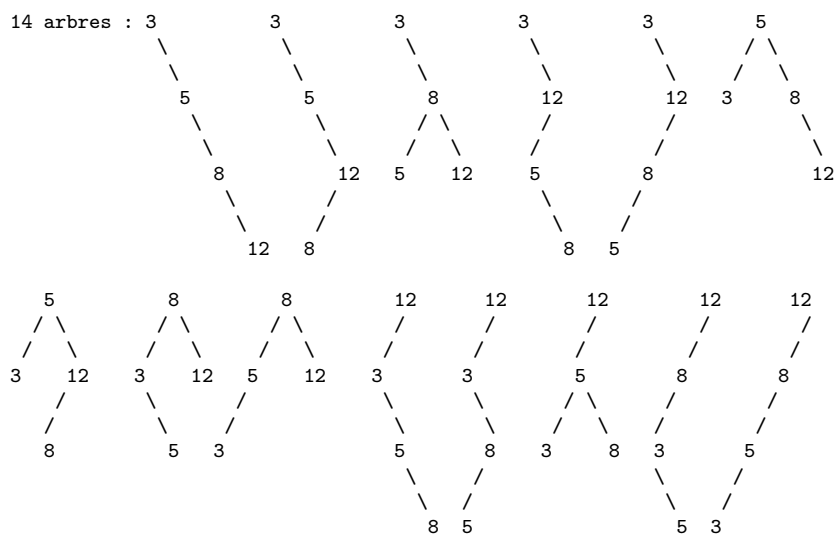


# TD n°1 - Correction

## Arbres binaires de recherche

### Exercice 1



### Exercice 2

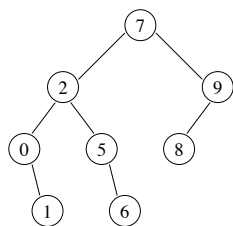


FIGURE 1 – ABR après insertions successives

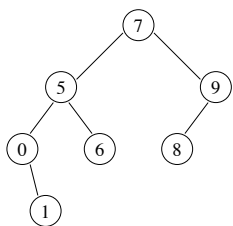


FIGURE 2 – ABR après suppression de 2 en le remplaçant par son successeur

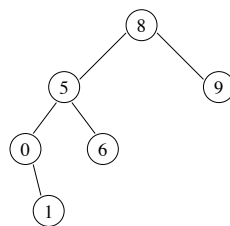


FIGURE 3 – ABR après suppression de 7 en le remplaçant par son successeur

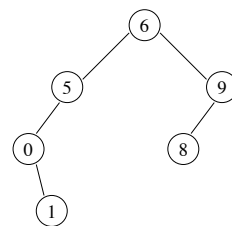


FIGURE 4 – ABR en remplaçant 7 par son prédécesseur

On remarque que l'on préserve mieux l'équilibre de l'arbre en alternant l'utilisation du prédécesseur et du successeur. Une solution est de tirer au hasard celui que l'on utilise.

**Exercice 3** Pour chaque séquence, il faut tenter de reconstituer le parcours dans l'ABR en vérifiant la propriété des ABR : "pour tout nœud  $x$ , les clés du sous-arbre gauche sont inférieures à la clé de  $x$ , les clés du sous-arbre droit sont supérieures à la clé de  $x$ ".

Pour les séquences 1, 2 et 4, c'est possible. Par contre pour les séquences 3 et 5, cela ne l'est pas.

Dans la séquence 3, la sous-séquence 911, 240, 912 est impossible, car 240 est nécessairement le fils gauche de 911 et donc 912 appartient au sous arbre gauche de 911, ce qui est impossible dans un ABR.

Dans la séquence 5, la sous-séquence 347, 621, 299 est impossible, car 621 est nécessairement le fils droit de 347 et donc 299 appartient au sous-arbre droit de 347, ce qui est impossible dans un ABR.

Finalement, en notant  $(x_i)$  la suite des valeurs parcourues, il suffit de vérifier que :

$$\text{Pour tout } i \left( (\forall k > i, x_k \leq x_i) \text{ OU } (\forall k > i, x_k \geq x_i) \right)$$

#### Exercice 4

- Pire ordre : 1 puis 2 puis 3... puis 1000. (Mais aussi : 1 puis 1000 puis 2 puis 999 puis 3... puis 500 puis 501)
- Meilleur ordre : on insère à chaque fois le milieu comme racine ; puis on recommence par récurrence. Plusieurs racines possibles. On prend par exemple 512 pour avoir une meilleure récursion. Plusieurs ordres possibles :
  - 512 256 128 64 32 16 8 4 2 1 3 6 5 7 12 10 9 11 14 13 15... (correspond à une création en parcours préfixe)
  - 512 256 768 128 384 640 896 64... (correspond à une création de haut en bas)

#### Exercice 5 Profondeur des arbres binaires.

1. Par récurrence sur  $h$ , on trouve que c'est  $\sum_{k=0}^h 2^k = 2^{h+1} - 1$ .
2. Puisque  $n \leq 2^{h+1} - 1$ , on a  $2^{h+1} \geq n + 1$  donc  $h \geq \lceil \log_2(n + 1) \rceil - 1$ .
3. Soit  $p(h)$  la profondeur moyenne d'un nœud de l'arbre binaire complet de hauteur  $h$ . Il est clair que  $p(h) \leq h$  puisque  $h$  est la profondeur maximale d'un nœud de l'arbre. Il suffit donc de montrer que  $p(h) \geq h - 1$ . On a  $p(0) = 0$  et  $p(1) = 2/3$  donc la propriété  $p(h) \geq h - 1$  est vérifiée pour  $h = 0$  et 1 (on aurait pu se passer du cas  $h = 1$ ).  
Par récurrence, supposons  $p(h) \geq h - 1$ . On remarque que sur les  $2^{h+2} - 1$  nœuds de l'arbre de hauteur  $h + 1$ ,  $2^{h+1} - 1$  d'entre eux forment l'arbre binaire complet de hauteur  $h$ , et les autres sont les feuilles à la profondeur  $h + 1$ . Donc

$$\begin{aligned} p(h+1) &= \frac{1}{2^{h+2} - 1} ((2^{h+1} - 1)p(h) + (h+1)2^{h+1}) \\ &\geq \frac{1}{2^{h+2} - 1} ((2^{h+1} - 1)(h-1) + (h+1)2^{h+1}) \\ &= \frac{1}{2^{h+2} - 1} (2^{h+1} \cdot 2h - (h-1)) = \frac{1}{2^{h+2} - 1} ((2^{h+2} - 1) \cdot h + 1) \geq h \end{aligned}$$

#### Exercice 6

**Somme(a) :**

- Si  $a$  est vide, renvoyer 0
- Renvoyer  $\text{Somme}(G(a)) + \text{Somme}(D(a)) + \text{Val}(a)$   
(Complexité linéaire)

**Maximum(a) :**

- Si  $a$  n'a pas de fils droit, renvoyer  $\text{val}(a)$
- Renvoyer  $\text{Maximum}(D(a))$   
(Complexité  $O(h)$ )

**Exercice 7** Si les nombres sont déjà triés, la construction de l'ABR prend  $1+2+3+\dots+n=O(n^2)$ . Ensuite un parcours les donne en  $O(n)$  dans tous les cas. Ce tri semble donc mauvais. Cependant :

- On verra en cours qu'au mieux et en moyenne l'insertion est en  $O(n \log n)$
- Les ABR équilibrés atteignent  $O(n \log n)$  dans tous les cas