

TD - Arbres binaires

Exercice 1. Nous connaissons déjà les arbres complets à gauche. On définit une propriété encore plus restrictive : un arbre binaire est dit *parfait* si tous ses étages sont complètement remplis, y compris le dernier !

1. Quelles valeurs peuvent être prises par le nombre total de noeuds d'un arbre binaire parfait ?
2. Écrire une fonction `parfait : int -> int` arbre prenant en entrée un entier naturel h et renvoyant un arbre binaire parfait à $2^h - 1$ noeuds internes, et dont les noeuds situés à la profondeur k portent l'étiquette $h - k$. *Indication : Le résultat attendu est bien moins impressionnant que sa description le laisse penser. Prenez le temps de faire un dessin pour vérifier proprement ce que vous devez renvoyer.*

Exercice 2. On s'intéresse tout d'abord à des arbres binaires *stricts*, c'est-à-dire que tout noeud interne doit posséder *exactement* 2 enfants.

1. Peut-on donner une définition inductive des arbres binaires stricts ? *Indication : repenser à la définition des arbres binaires. Quel élément de cette définition permet de créer des arbres non stricts ? Quelle modification peut-on alors apporter ?*
2. Soit A est un arbre binaire strict possédant n_i noeuds internes et f feuilles, montrer qu'alors on a $f = n_i + 1$.
3. Soit $k \in \mathbb{N}$. On appelle maintenant *arbre k-aire strict* un arbre dont tous les noeuds internes ont exactement k enfants. En s'inspirant du résultat précédent, trouver une relation entre le nombre $n_i(T)$ de noeuds internes d'un arbre k -aire strict T et son nombre $f(T)$ de feuilles, et la prouver. *Indication : Pour trouver la bonne formule, le mieux est encore d'essayer sur quelques exemples...*

Exercice 3. Parmi ces suites de nombres, lesquelles peuvent représenter la liste des étiquettes parcourues dans un chemin depuis la racine d'un ABR vers une de ses feuilles ? Lesquelles ne peuvent pas ? Justifier.

1. 2,252,401,398,330,344,397,363
2. 924,220,911,244,898,258,362,363
3. 925,202,911,240,912,245,363
4. 2,399,387,219,266,382,381,278,363
5. 935,278,347,621,299,392,358,363

Exercice 4. Soit t un ABR, et x_0 une feuille de l'ABR. Notons B l'ensemble des noeuds qui se trouvent sur le chemin direct de la racine de t à sa feuille x_0 . On répartit les noeuds de t dans trois ensembles A , B et C tels que A est l'ensemble des noeuds qui se trouvent à gauche du chemin B dans l'arbre, et C est l'ensemble des noeuds qui se trouvent à droite du chemin B dans t . Considérons la propriété P sur t : "Pour toutes étiquettes a, b et c de t , si $a \in A, b \in B$ et $c \in C$ alors on a $a \leq b \leq c$."

La propriété P est-elle vraie ? Si oui, la prouver. Si non, donner un contre-exemple le plus petit possible.

Exercice 5. Pour chacune de ces questions, apporter une preuve ou un contre-exemple.

1. L'ordre des opérations d'insertion dans un ABR a-t-il une importance ? Autrement dit, si l'on insère un élément x puis un élément y dans un ABR, obtient-on forcément le même arbre que si on insère d'abord l'élément y puis l'élément x dans l'arbre ?
2. L'ordre des opérations de suppression dans un ABR a-t-il une importance ? Autrement dit, si l'on supprime un élément x puis un élément y dans un ABR, obtient-on forcément le même arbre que si on supprime d'abord l'élément y puis l'élément x dans l'arbre ?

Exercice 6.

1. Le tableau $\{23, 17, 14, 6, 13, 10, 1, 5, 7, 12\}$ représente-t-il un tas-max ?
2. Où se trouve le plus grand élément dans un tas-min, en supposant que les éléments sont deux à deux distincts ?
3. Un tableau trié représente-t-il un tas ? Si oui, de quel tas s'agit-il (min ou max) ? Si non, le justifier rapidement.
4. La réciproque est-elle vraie ? Si oui, le prouver. Si non, donner un contre-exemple.

Exercice 7. Soit $k \geq 2$ un entier. Un tas k -aire ressemble à un tas binaire, à ceci près qu'un noeud qui n'est pas une feuille a k enfants au lieu de 2 (à une exception potentielle près).

1. Comment pourrait-on représenter un tas k -aire dans un tableau ?
2. À quels indices sont les fils du noeuds d'indice i ? À quel indice est le parent du noeud d'indice i ?
3. Quelle est la hauteur d'un tas k -aire à n éléments en fonction de n et k ?
4. Donner une implémentation efficace de l'extraction du minimum tas-min k -aire. Analyser son temps d'exécution en fonction de k et n .
5. Donner une implémentation efficace de l'insertion pour un tas min k -aire. Analyser son temps d'exécution en fonction de k et n .

Exercice 8. On dispose de k listes que l'on suppose initialement triées. On souhaite former une unique liste triée contenant les éléments de toutes les listes initiales (autrement dit, on souhaite réaliser une fusion de listes comme dans un tri fusion, mais en réunissant k listes plutôt que 2).

Donner un algorithme en temps $O(n \cdot \log(k))$ qui réalise cette fusion de k listes, où n désigne le nombre total d'éléments dans la réunion de ces listes.

Indication : La difficulté est d'atteindre la complexité demandée. Servez-vous des outils du chapitre!