

ARBRES BINAIRES DE RECHERCHE

Sauf indication contraire, les arbres sont binaires et homogènes de type :

type 'a arbre = V | S of ('a * 'a arbre * 'a arbre).

La taille d'un arbre binaire est le nombre de ses sommets.

1)

- Écrivez une fonction qui calcule le maximum d'un ABR. Quelle est sa complexité en fonction de h , hauteur de l'arbre ? en fonction de n , taille de l'arbre ?
- Écrivez une fonction qui supprime un élément d'un ABR et qui retourne un ABR. Estimez sa complexité.
- Écrivez une fonction qui fusionne deux ABR en un seul ABR.

2)

- Écrivez une fonction qui coupe un ABR a en deux ABR b, c en fonction d'un objet x : l'ABR b contient tous les éléments de l'ABR a qui sont strictement inférieurs à x et c les éléments qui sont strictement supérieurs à x .
- Faites de même en plaçant dans c les éléments qui sont supérieurs ou égaux à x .

3) Écrivez une fonction qui détermine le prédécesseur d'un objet dans un ABR, c'est-à-dire le plus grand élément de l'ABR inférieur ou égal à l'objet. Faites de même avec le successeur.

4) [Transformation d'un tableau croissant en un ABR]

Soit t un tableau d'éléments rangés dans l'ordre croissant. Écrivez une fonction qui construit un ABR qui contient les éléments du tableau en utilisant la méthode « diviser pour régner ». Calculez sa complexité en fonction de la longueur n du tableau. Quelle est la hauteur de l'ABR en fonction de n ?

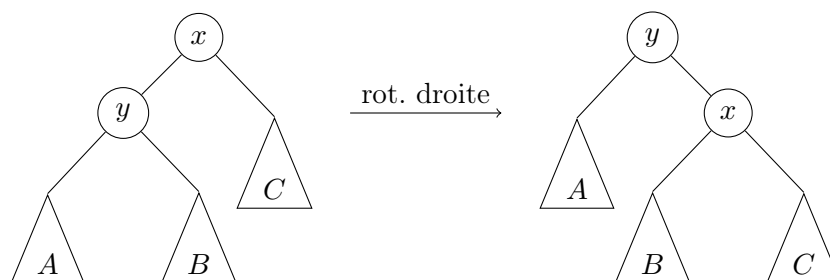
On dit qu'un arbre binaire est équilibré quand son équilibre e , égal à la différence entre la hauteur de son fils gauche et de son fils droit, satisfait l'inégalité $|e| \leq 1$ et quand ses fils sont eux-mêmes équilibrés. Par convention, l'équilibre de l'arbre vide vaut 0 et il est équilibré.

5)

- Justifiez que l'ABR construit dans l'exercice précédent est équilibré.
- Donnez le principe d'un algorithme qui ajoute un élément à un ABR en le gardant équilibré et sa complexité en fonction de la taille de l'ABR. Est-il acceptable ? Quel est son défaut ?
- Donnez de même le principe d'un algorithme qui construit par ajouts successifs un ABR à partir d'une liste d'éléments. Donnez sa complexité en fonction de la longueur de la liste. Est-ce une bonne idée de procéder de cette façon ?

Deux opérations sont dans certains cas utilisées pour rééquilibrer un arbre légèrement déséquilibré.

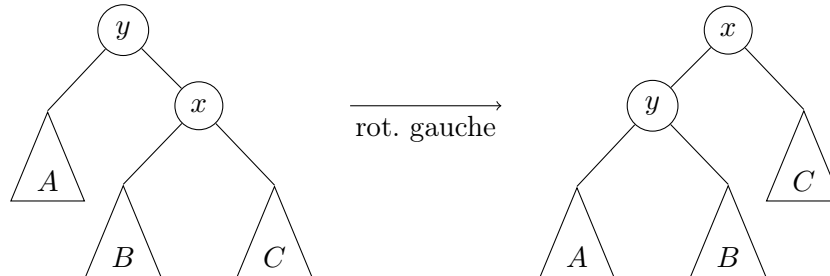
6) [Étude de la rotation droite]



- Écrivez une fonction qui réalise la rotation droite d'un arbre.
- Justifiez que la rotation droite d'un ABR donne un ABR.
- Montrez que la hauteur de l'arbre après rotation diminue de 1 si $h(A) > \max(h(B), h(C))$, augmente de 1 si $h(C) > \max(h(A), h(B))$ ou ne varie pas dans les autres cas.

- d) Si on note $e(x)$ l'équilibre du nœud portant l'étiquette x dans l'arbre initial, et $e'(x)$ celui dans l'arbre après rotation droite, montrez que $e'(x) = e(x) - 1 - \max(0, e(y))$ et $e'(y) = \begin{cases} e(y) - 1 & \text{si } e'(x) \geq 0 \\ e(x) - 2 + \min(0, e(y)) & \text{sinon} \end{cases}$.
- e) On note Δh la variation de hauteur entre les deux arbres. Montrez que si $e(x) \leq 0$, alors $\Delta h = 1$, sinon si $e'(y) \geq 0$ alors $\Delta h = -1$, sinon $\Delta h = 0$.

7) Faites de même avec la rotation gauche



8) [Enracinement]

Écrivez une fonction `enraciner a x` de type `'a arbre -> 'a -> 'a arbre`, qui prend en paramètres un ABR `a` et une étiquette `x` qui apparaît dans l'ABR et qui construit l'ABR avec les mêmes éléments et une racine d'étiquette `x`. Quelle est sa complexité ?