

## ARBRES BINAIRES

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) Pour se remettre en jambes.

- a) Écrivez des fonctions qui calculent la taille d'un arbre, sa hauteur.
- b) Écrivez des fonctions qui calculent les listes des feuilles ou des nœuds d'un arbre (dans un ordre quelconque).

2) On suppose que les étiquettes sont toutes distinctes. Écrivez une fonction qui donne la liste des ancêtres d'un sommet, c'est-à-dire tous les sommets qui se trouvent sur le chemin menant de la racine à un sommet

3) On appelle équilibre d'un arbre  $(a, (B', B''))$  la différence  $h(B') - h(B'')$ , où  $h$  est la fonction hauteur sur les arbres binaires. Par convention, l'équilibre de l'arbre vide vaut 0. Un arbre est dit équilibré quand son équilibre  $e$  vérifie  $|e| \leq 1$  et ses fils sont eux-mêmes équilibrés.

- a) Écrivez une fonction qui vérifie si un arbre est équilibré ou non. Quelle est sa complexité en fonction de la taille de l'arbre ?

- b) Montrez que si un arbre équilibré possède  $n$  sommets et est de hauteur  $h$ , alors  $h \leq \frac{3}{2} \log(n + 1)$ .

4) Équilibrer un arbre n'est pas toujours évident, surtout si on doit respecter d'autres contraintes. On propose ici une façon rudimentaire de construire des arbres équilibrés par insertion successives d'éléments.

Pour cela, on modifie le type **arbre** : **type** 'a arbre = V | S of 'a \* 'a arbre \* 'a arbre \* int \* int. Les deux entiers ajoutés dans l'étiquette du sommet sont les hauteurs de ses fils gauche et droit.

- a) Si on doit ajouter une feuille dans un arbre équilibré de ce type, on choisit de l'ajouter dans le fils le moins haut (en cas d'égalité des hauteurs, on choisit arbitrairement l'un des deux). Montrez que l'arbre obtenu reste équilibré.
- b) Écrivez une fonction **ajouter**  $x$  **a** de type 'a -> 'a arbre -> 'a arbre qui prend en paramètres une étiquette et un arbre équilibré et qui construit un nouvel arbre équilibré en ajoutant la feuille d'étiquette  $x$ . Donnez sa complexité en fonction de  $h$ , hauteur de l'arbre, ou de  $n$ , taille de l'arbre.
- c) Écrivez une fonction qui prend une liste d'objets et qui construit un arbre équilibré les contenant. Calculez sa complexité en fonction du nombre d'éléments de la liste.

5) On remplit un arbre successivement de la façon suivante : partant de l'arbre vide, on ajoute toujours le nouvel élément à droite, puis on échange les fils droit et gauche. Montrez qu'on obtient un arbre équilibré.

6) [Nombre de Strahler] Un fleuve coule depuis sa source vers son embouchure : on peut le voir comme un arbre binaire entier dont la racine est l'embouchure et chaque branche comme deux rivières confluentes en un sommet, les feuilles représentant les sources. Pour différencier les bras et les classer par ordre d'importance, on leur attribue un entier appelé nombre de Strahler : pour simplifier, on donne le nombre au sommet qui débute le bras. À titre d'exemple, l'Amazone a un nombre de Strahler de 12, le Nil de 10 et la Loire de 8 ; ce nombre mesure l'importance du fleuve et de la densité de son bassin versant.

Récursivement,

- une feuille a un nombre de Strahler égal à 1 ;
  - si un nœud a deux fils de nombres de Strahler  $a$  et  $b$ , alors il a lui-même pour nombre  $a + 1$  si  $a = b$  et  $\max(a, b)$  si  $a \neq b$ .
- a) Déterminez le nombre de Strahler d'un arbre complet de hauteur  $h$  (complet signifiant entièrement rempli).
  - b) Écrivez une fonction qui calcule le nombre de Strahler d'un arbre binaire entier.

## 7) [Linéarisation d'un arbre]

À tout arbre binaire homogène, on associe la liste de ses éléments parcourus dans l'ordre suffixe, appelée liste suffixe de l'arbre.

- a) Écrivez une fonction `liste_suffixe a` qui calcule la liste suffixe des éléments d'un arbre `a`. Donnez un exemple de deux arbres distincts qui ont même liste suffixe.

Lorsqu'on crée la liste suffixe des éléments, on perd la structure de l'arbre.

On peut faire mieux : au lieu de ranger dans la liste uniquement la valeur du sommet, on va aussi enregistrer si c'est une feuille ou un nœud (interne) grâce au type suivant

```
type 'a contenu = F of 'a | N of 'a;;
```

- a) Modifiez la fonction précédente pour qu'elle enregistre l'information supplémentaire dans la liste : la liste est appelée liste suffixe complète de l'arbre.
- b) Écrivez la fonction réciproque, qui construit l'arbre associé à une liste suffixe complète. Si la liste n'est pas une liste suffixe complète, on donne un message d'erreur.

8) Montrez que si on connaît les deux listes des sommets d'un arbre binaire parcouru en profondeur préfixe et en profondeur infixé, alors on peut reconstituer l'arbre.