Option Informatique en Spé MP et MP*
Arbres binaires, arbres binaires de recherche (en abrégé : abr)

Question 1 $\star$ Décrivez un algorithme calculant la profondeur *minimale* d'une feuille, dans un arbre binaire de recherche. Rédigez en Caml une fonction implémentant cet algorithme.

Question 2 $\star$ **Rotations**. Soit $a = N(N(g', y, g''), x, d)$ un arbre binaire non réduit à une feuille, et dont le sous-arbre gauche n'est pas non plus réduit à une feuille. L'image de cet arbre par la *rotation droite* autour de sa racine est l'arbre binaire $a' = N(g', y, N(g'', x, d))$. Faites une figure. Montrez que si a est un abr, il en est de même de a'. Définissez la rotation gauche. Rédigez en Caml des fonctions qui réalisent ces transformations.

Question 3 $\star$ On suppose défini le type `arbre` suivant :

```
type 'a arbre = F | N of 'a * 'a arbre * 'a arbre;;
```

Rédigez en Caml une fonction :

```
maptree : ('a -> 'b) -> 'a arbre -> 'b arbre
```

spécifiée comme suit : `maptree f a` construit un nouvel arbre, obtenu par application de la fonction f à chaque nœud de l'arbre a. Donnez une CNS portant sur f pour que l'image par f d'un abr soit un abr.

Question 4 $\star\star$ Décrivez un algorithme calculant la plus petite clé d'un abr t strictement supérieure à une valeur x donnée. Rédigez en Caml une fonction implémentant cet algorithme.

Question 5 $\star\star$ Soit t un arbre binaire de recherche, obtenu à partir d'un arbre vide par une suite d'insertions et de suppressions, réalisées avec les algorithmes exposés en cours. On supprime deux nœuds de t. Le résultat dépend-il de l'ordre dans lequel on a effectué ces suppressions ?

Question 6 $\star$ **Scission d'un arbre binaire de recherche**. Soient $x \in \mathbb{N}$ et A un arbre binaire de recherche, dont les nœuds sont étiquetés par des entiers. Décrivez un algorithme réalisant la *scission* (ou *coupure*) de A selon x : le résultat attendu est un couple (A_1, A_2) d'arbres binaires de recherche tel que tout nœud de A_1 (resp. A_2) est inférieur ou égal (resp. strictement supérieur) à x, et l'ensemble des nœuds de A est la réunion des nœuds de A_1 et A_2.

Question 7 **Fusion de deux arbres binaires de recherche**. Décrivez un algorithme réalisant la *fusion* de deux arbres binaires de recherche. Vous commencerez par donner une spécification précise.

Question 8 **Équivalence de deux arbres binaires de recherche** (FGS 10.8). Nous dirons que deux arbres binaires de recherche sont *équivalents* s'ils contiennent exactement les mêmes éléments. Décrivez un algorithme testant l'équivalence de deux abr.

Question 9 **Préfixes d'un arbre binaire**. On dit qu'un arbre binaire t est un *préfixe* d'un arbre binaire t' si t est l'arbre vide, ou si $t = N(t_1, r, t_2)$, $t' = N(t'_1, r, t'_2)$, t_1 (resp. t_2) étant un préfixe de t'_1 (resp. de t'_2). Montrez que la relation «est préfixe de» est un ordre partiel sur les arbres binaires. Montrez que tout préfixe d'un abr est un abr. Rédigez en Caml une fonction testant si un arbre binaire est préfixe d'un autre.

Question 10 **Suffixes d'un arbre binaire**. On dit qu'un arbre binaire t est un *suffixe* d'un arbre binaire t' si $t = t'$, ou si $t' = N(t'_1, r, t'_2)$, t étant un suffixe de t'_1 ou de t'_2. Montrez que la relation «est suffixe de» est un ordre partiel. Montrez que tout suffixe d'un abr est un abr. Rédigez en Caml une fonction testant si un arbre binaire est suffixe d'un autre.

Question 11 **Facteurs d'un arbre binaire**. On dit qu'un arbre binaire t est un **sous-arbre** d'un arbre t' s'il existe une suite finie d'arbres $t_0 = t, t_1, \ldots, t_n = t'$ telle que, pour tout i $(0 \leqslant i < n)$, t_i soit préfixe ou suffixe de t_{i+1}. Montrez que la relation «est sous-arbre de» est un ordre partiel. Montrez que tout sous-arbre d'un abr est un abr. Rédigez en Caml une fonction testant si un arbre binaire est sous-arbre d'un autre.

Question 12 **Recherche adaptative dans un abr** (FGS 10.13). Après chaque recherche réussie dans un abr, on ramène l'élément recherché (et trouvé) à la racine par une suite de rotations gauches ou droites. Soit $(e_i)_{1 \leqslant i \leqslant n}$ l'énumération des éléments d'un tel arbre, par ordre croissant. Soient e_i et e_j deux éléments d'un tel abr ; on suppose $e_i < e_j$. Montrez que e_i est un ancêtre de e_j ssi l'une des deux conditions suivantes est satisfaite :

- ou bien e_i était au départ un ancêtre de e_j, et il n'y a eu aucune recherche portant sur $e_{i+1}..e_j$;
- ou bien il n'y a eu aucune recherche portant sur $e_{i+1}..e_j$ depuis la dernière recherche de e_i.

FIN