TP n° 4 – Structures arborescentes

esimon@esimon.fr thomas.h@ens.fr
thomash.fr/teaching/

1 Oups !

Question 1. Yorel Reivax définit le type d'un arbre binaire comme suit :

```
type 'a arbre = Noeud of 'a * 'a arbre * 'a arbre ;;
```

Il passe plusieurs jours sur son clavier à écrire son premier arbre, il n'en peut plus, aidez le !

2 Introduction aux arbres binaires de recherche

Un arbre binaire de recherche (ABR) est un arbre qui satisfait la propriété : toutes les clefs du sous-arbre gauche d'un nœud sont inférieures à la clef du nœud courant et toutes les clefs du sous-arbre droit d'un nœud sont supérieure à la clef du nœud courant. En voici un exemple [1] :

```
let perdu = Noeud(15,
  Noeud(4, Vide, Noeud(8, Vide, Vide)),
  Noeud(23, Noeud(16, Vide, Vide), Noeud(42, Vide, Vide)));;
```

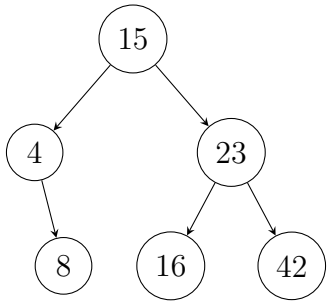

FIGURE 1 – L'ABR `perdu` donné en exemple.

Question 2. Écrire les fonctions `arbre_taille : 'a arbre -> int` et `arbre_hauteur : 'a arbre -> int` qui renvoient respectivement le nombre de nœuds et la hauteur d'un arbre [2].

Question 3. Écrire une fonction `parcours_infixe : 'a arbre -> 'a list` qui renvoie la liste des clefs d'un arbre selon l'ordre infixe (selon l'ordre : sous-arbre gauche, nœud courrant puis sous-arbre droit.)

Question 4. Écrire une fonction `chercher_abr : 'a arbre -> 'a -> bool` qui cherche si une clef est présente dans un ABR.

Question 5. Écrire une fonction `verifier_abr : 'a arbre -> bool` qui vérifie si un arbre donné en argument est bien un ABR.

Question 6. Écrire une fonction `ajouter_abr : 'a arbre -> 'a -> 'a arbre` qui étant donné un ABR et une valeur renvoie un ABR avec cette nouvelle valeur ajoutée en feuille.

1. Évidement, vous avez réfléchi à la question 1 sans lire la suite du sujet.
2. Si vous voulez faire du récursif terminal, sachez que vous devez maintenir un « accumulateur temporaire » sous forme de liste de sous-arbres restant à visiter.

Question 7. À partir des questions précédentes, déduire une fonction `tri_abr : 'a list -> 'a list` qui trie une liste. Quelle est sa complexité ?

Question 8. * Écrire une fonction `supprimer_abr : 'a arbre -> 'a -> 'a arbre` qui supprime une valeur d'un ABR (indice : regardez les extrema des sous-arbres.)

3 Tas-max

Un tas-max est un arbre binaire complet dans lequel la clef d'un nœud est toujours inférieure ou égale à celle de son parent. Il est d'ordinaire de représenter ces tas binaires par un tableau, l'indice de chaque élément dans le tableau est son ordre d'apparition dans le parcours en largeur de l'arbre. L'arbre étant complet, il est possible d'accéder directement au père en divisant l'indice par deux ($\lfloor \frac{i-1}{2} \rfloor$) ou d'accéder aux fils en le multipliant par deux ($2i+1$ et $2i+2$).

La taille du tas étant appelée à être modifiée, celui-ci est représenté par un couple `(t, n)` où `t` est un tableau de la taille maximale dont on aura besoin et `n` la taille réelle du tas.

```
type 'a tas == 'a vect * int ref ;;
```

Question 9. Écrire une fonction `tamiser : 'a tas -> int -> echange` qui étant donné un tas et un nœud d'indice i force la propriété de tas-max avec ses fils $2i+1$ et $2i+2$ en faisant un échange si nécessaire, cette fonction renverra le type d'échange effectué.

Question 10. Écrire une fonction `tamiser_branche : 'a tas -> int -> unit` qui étant donné un nœud i dont les deux sous arbres fils sont des tas-max, force la propriété de tas-max sur le sous arbre enraciné en i.

Question 11. Écrire une fonction `entasser : 'a vect -> 'a tas` qui transforme un tableau quelconque en tas-max.

Question 12. Écrire une fonction `ajouter_tas : 'a tas -> 'a -> unit` qui ajoute un élément à un tas-max (commencez par mettre l'élément à la fin du tas-max puis forcez la propriété).

Question 13. Écrire une fonction `extraire_tas : 'a tas -> 'a` qui supprime l'élément maximum d'un tas-max et le renvoie.

Question 14. En déduire une fonction `tri_tas : 'a list -> 'a list`. Quelle est sa complexité ?

Question 15. * Implémenter une version fonctionnelle pure des fonctions précédentes[3].

Question 16. * Imaginer un moyen de combiner un ABR avec un tas-max pour obtenir de meilleures performances sur les fonctions de l'exercice 2.

3. La référence en matière d'implémentation fonctionnelle des structures des données est *Purely functional data structure* de Chris Okasaki qui est à la fois une thèse et un livre basé sur celle-ci.