

TP n°5 : Types structurés : les Arbres

Q1 La chauffe

Réécrivez dans un module `Tree.hs` les fonctions suivantes sans utiliser de schémas de programmes :

- Hauteur d'un arbre
- Nombre de feuilles d'un arbre
- Création de la liste des éléments d'un arbre
 - * dans l'ordre préfixé
 - * dans l'ordre symétrique
 - * dans l'ordre postfixé
- Afficher l'arbre de la figure `FIGURE 1` sous la forme :

```
"1"  
" |__2"  
"   |__4"  
"   |__5"  
"       |__7"  
"       |__8"  
" |__3"  
"   |__6"  
"       |__9"
```

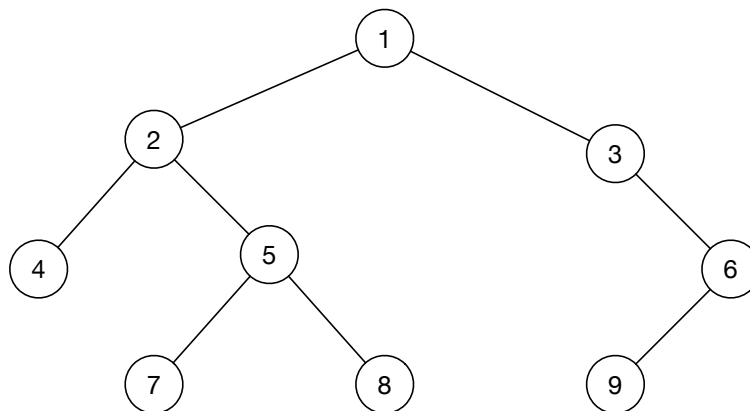


FIGURE 1 – Arbre binaire

Q2 Tri par arbre

Dans cet exercice on se sert de deux types d'arbres particuliers :

les arbres binaires de recherche (ABR) : la valeur de chaque noeud est \geq à celle de tous les noeuds de son sous-arbre gauche, comme dans la FIGURE 2

les tas : la valeur de chaque noeud fils est \geq à celle de son père, comme dans la FIGURE 3

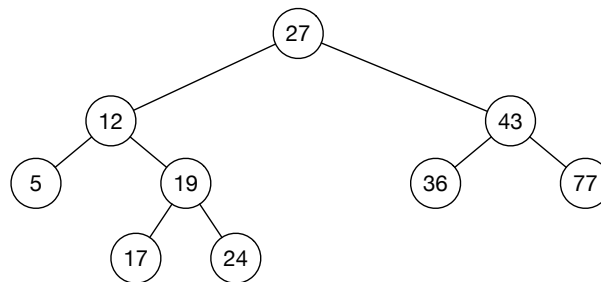


FIGURE 2 – Arbre Binaire de Recherche

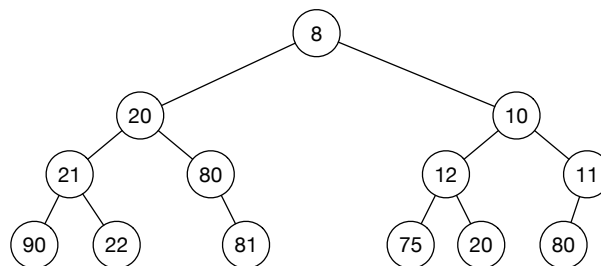


FIGURE 3 – Arbre en Tas

- (a) Écrire une fonction `bien_forme` qui teste si un arbre est un ABR.
- (b) Écrire une fonction `insere` qui ajoute un élément à la bonne place dans un ABR
- (c) Écrire une fonction `liste_vers_arbre` qui transforme une liste d'éléments en un ABR
- (d) Écrire une fonction `arbre_vers_liste` qui transforme un ABR en une liste triée dans l'ordre croissant
- (e) Écrire une fonction `tri` qui trie une liste en utilisant un ABR.
- (f) **Refaites les mêmes questions pour un tas**