Algorithmique et Structures de données

Feuille 7 : Arbre Binaire de Recherche

Exercice 7.1

Dessiner des arbres binaires de recherche de hauteur $2, 3, 4, 5$, et 6 pour le même ensemble de clés $\{1, 4, 5, 10, 16, 17, 21\}$.

Exercice 7.2

On suppose que les entiers compris entre 1 et 1000 sont disposés dans un arbre binaire de recherche, et on souhaite retrouver le nombre 363. Parmi les séquences suivantes, lesquelles ne pourraient pas être la séquence de noeuds parcourus ?

1. $2, 252, 401, 398, 330, 344, 397, 363$.
2. $924, 220, 911, 244, 898, 258, 362, 363$
3. $925, 202, 911, 240, 912, 245, 363$
4. $2, 399, 387, 219, 266, 382, 381, 278, 363$
5. $935, 278, 347, 621, 299, 392, 358, 363$

Exercice 7.3

1. Ajouter successivement à l'arbre binaire dessiné sur la Fig. 1 les valeurs : $9, 15, 16, 7, 23, 5$
2. Dessiner tous les arbres binaires de recherche valués sur $V = \{10, 20, 30, 40\}$ ayant 20 comme valeur à la racine.
3. Donner le mot infixe obtenu en écrivant les valeurs des sommets dans l'ordre de parcours infixe.
4. Ecrire une fonction d'ajout d'une valeur n à un arbre binaire de recherche.
5. Ecrire deux fonctions qui retournent respectivement la plus petite et la plus grande valeur contenue dans un arbre binaire de recherche. Ecrire les fonctions qui suppriment ces deux éléments.
6. Ecrire une fonction de recherche d'un élément donné dans un arbre binaire de recherche.
7. Ecrire une fonction qui vérifie si un arbre binaire est un arbre binaire de recherche.

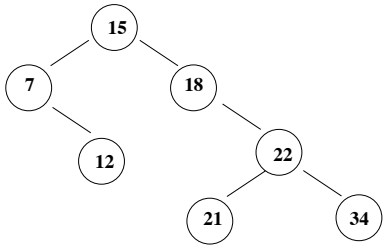

FIGURE 1 – Exemple d'ABR

Exercice 7.4
Soit l'arbre binaire de recherche donné sur la Fig. 2.

1. Dessiner l'arbre après la suppression du noeud `13`.
2. L'opération suppression est-elle "commutative" au sens où la suppression de `x` puis de `y` dans un arbre binaire de recherche produit le même arbre que la suppression de `y` puis de `x`.
   Si oui dire pourquoi, sinon donner un contre exemple.

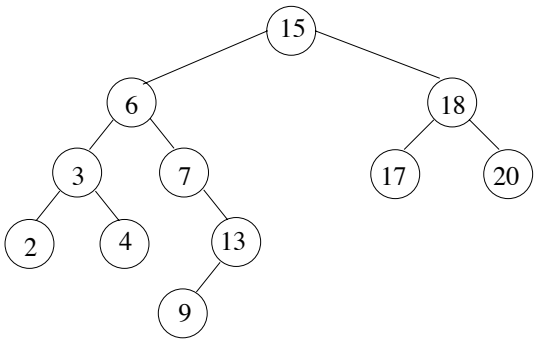

FIGURE 2 – ABR

Problème récurrent (notre fil d'ariane)
Exercice 7.5 Gestion d'une piste d'atterrissage des avions

Un avion est un enregistrement contenant :
– l'indicatif (6 caractères)
– la destination (30 caractères)
– l'autonomie résiduelle de carburant comptée en heures de vol (entier)
– deux booléens indiquant s'il y a un pirate à bord et s'il y a le feu.

1. Définir les structures de données nécessaires.
2. Ecrire la fonction `Priorité` ainsi que la gestion complète de la piste.
3. Envisager le cas de suppression d'un élément quelconque de la file lorsque le pirate a mis sa menace de détournement à exécution.

Quelles sont les notions que vous venez de voir qui peuvent permettre d'amorcer le fil d'ariane ?

Annexe A : **Arbres binaires de recherche** *ABR*

On utilise les primitives des arbres binaires.
De plus, les primitives `ajouter` et `supprimer` permettent de faire évoluer un ABR.

```
fonction ajouter(ref x: sommet, val e: objet): vide;
fonction supprimer(ref x: sommet): booleen;
```

Annexe B **Type abstrait** *arbreBinaire*

```
arbreBinaire= curseur;
sommet= curseur;
```

- **Création**
  ```
  fonction creerArbreBinaire(val Racine:objet):sommet;
  fonction detruireArbreBinaire(ref a: arbreBinaire d'objet):vide;
  ```
- **Accès**
  ```
  fonction getValeur(val S:sommet):objet;
  fonction filsGauche(val S:sommet):sommet;
  fonction filsDroit(val S:sommet):sommet;
  fonction pere(val S:sommet):sommet;
  ```
- **Modification**
  ```
  fonction setValeur(ref S:sommet, val x:objet):vide;
  fonction ajouterFilsGauche(ref S:sommet, val x:objet):vide;
  fonction ajouterFilsDroit(ref S:sommet, val x:objet):vide;
  fonction supprimerFilsGauche(ref S:sommet):vide;
  fonction supprimerFilsDroit(ref S:sommet):vide;
  fonction detruireSommet(ref S:sommet):vide;
  ```

Annexe C **Implémentation du type abstrait** *arbreBinaire*

```
cellule=structure
         info:objet;
         gauche: sommet;
         droit: sommet;
         pere: sommet;
       finstructure
curseur=^cellule;

arbreBinaire= curseur;
sommet= curseur;
```