Travaux Dirigés d'algorithmique n°7

Cours d'algorithmique

—Licence CFA / Troisième Année—

Les arbres binaires

Cette séance de Travaux dirigée est consacrée à l'étude du type de donnée arbre binaire.

▸ Exercice 1. (Arbres binaires)

1. Implanter la spécification ABin en utilisant les déclarations suivantes :

```
typedef ... arbre;
arbre empty();
int is_empty(const arbre);
arbre sad(arbre);
arbre sag(arbre);
```

2. Écrire les fonctions `taille` et `hauteur`

On dit qu'un arbre est *filiforme* si chaque noeud a au moins un fils vide :

3. Dessiner tous les arbres filiformes de taille 1, 2 et 3
4. Combien y a-t-il d'arbres filiformes de taille n ?
5. Écrire une fonction `est_filiforme`.
6. Montrer que pour tout arbre filiforme t on a l'égalité

$$\texttt{taille}(t) = \texttt{hauteur}(t)$$

On dit qu'un arbre est un *peigne gauche* si le fils droit de chaque noeud est vide. On défini de même les peignes droits.

7. Écrire une fonction `peigne` qui retourne le peigne gauche de taille n donné.
8. Exécuter `est_filiforme`(`peigne`(3)). Qu'en pensez vous ?

▸ Exercice 2. (Chemins dans les arbres binaires)

On représente un chemin dans un arbre par un tableau de booléen (avec sa taille).

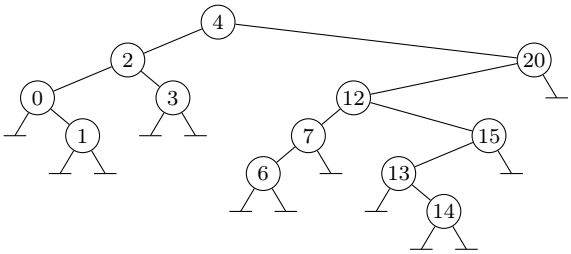
1. Écrire une fonction qui étant donnés un arbre et un chemin retourne le sous arbre correspondant.
2. Écrire une fonction qui étant donnés un arbre et un numéro retourne le sous arbre correspondant.

On code les arbres binaires avec des tableaux de booléen : La case i contient VRAI si le noeud de numéro i est dans l'arbre et FAUX sinon.

3. Donner la liste des arbres binaire à $1, 2, 3$ et 4 noeuds ainsi que leur codage par vecteurs booléens.
4. Comment vérifier si un tableau de booléen correspond bien à un arbre ?
5. Implanter la spécification ABin en utilisant le codage booléen.
6. Que pensez vous de l'utilisation de la mémoire ?
7. Dessiner l'arbre correspondant à une suite de k VRAI suivi par des FAUX.

▸ **Exercice 3. (Parcours d'arbre)**

1. Donner l'ordre de parcours des noeuds de l'arbre suivant, pour l'ordre préfixe. Quel est l'affichage si l'on affiche les étiquettes dans cet ordre ?



2. Même question pour les ordres infixe, et postfixe.
3. Même question pour un parcours en profondeur ?
4. Écrire un algorithme qui étant donné un arbre retourne la liste des étiquettes de cet arbre dans l'ordre de parcours préfixe.

▸ **Exercice 4. (Arbres binaires de recherches, suppression)**

Pour chacun des algorithmes suivants, on donnera la complexité.

1. Écrire un algorithme qui étant donné un ABR non vide retourne la plus petite valeur de cet ABR.
2. Écrire un algorithme qui étant donné un ABR non vide supprime et retourne la plus petite valeur de cet ABR.
3. Écrire un algorithme qui étant donné un ABR a supprime la racine de cet ABR. Si a et les deux sous arbres gauche et droit sont tous non vide, on remplacera la racine par le plus petit noeud du sous arbre gauche de a. Pourquoi ce choix est-il correct ?
4. Écrire un algorithme qui étant donné un élément e et un arbre a supprime e de a s'il est dans a et sinon laisse a inchangé.

▸ **Exercice 5. (Partitions d'arbres binaires de recherches)**

1. Écrire un algorithme qui étant données a un ABR et e un élément retourne un ABR a' qui ne contient que les éléments de a inférieur à e. Contrainte : L'algorithme ne doit parcourir qu'un seul chemin dans l'ABR sans comparer tout les noeuds à e.