

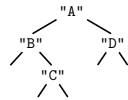
DIU Enseigner l'informatique au lycée  
mercredi 8 juillet (matin)

## Arbres binaires

**Exercice 1** Dessiner tous les arbres binaires ayant respectivement 3 et 4 nœuds. □

**Exercice 2** Sachant qu'il y a 1 arbre binaire vide, 1 arbre binaire contenant 1 nœud, 2 arbres binaires contenant 2 nœuds, 5 arbres binaires contenant 3 nœuds et 14 arbres binaires contenant 4 nœuds, calculer le nombre d'arbres binaires contenant 5 nœuds. On ne cherchera pas à les construire tous, mais seulement à les dénombrer. □

**Exercice 3** Écrire une fonction `affiche(a)` qui imprime un arbre sous la forme suivante : pour un arbre vide, on n'imprime rien ; pour un nœud, on imprime une parenthèse ouvrante, son sous-arbre gauche (récursivement), sa valeur, son sous-arbre droit (récursivement), puis enfin une parenthèse fermante. Ainsi, pour l'arbre



on doit afficher `((B(C))A(D))`. □

**Exercice 4** Écrire une fonction `peigne_gauche(h)` qui reçoit en argument un entier `h` supérieur ou égal à zéro et renvoie un peigne de hauteur `h` où chaque nœud a un sous-arbre droit qui est vide. □

**Exercice 5** Donner cinq arbres de taille 3, différents, dont les nœuds contiennent les valeurs 1, 2, 3 et pour lesquels la fonction `parcours_infixe` affiche à chaque fois

1  
2  
3

dans cet ordre. □

## Arbres binaires de recherche

**Exercice 6** Donner tous les ABR formés de trois nœuds et contenant les entiers 1, 2 et 3. □

**Exercice 7** Dans un ABR, où se trouve le plus petit élément ? En déduire une fonction `minimum(a)` qui renvoie le plus petit élément de l'ABR `a`. Si l'arbre `a` est vide, alors cette fonction renvoie `None`. □

**Exercice 8** Écrire une variante de la fonction `ajoute` qui n'ajoute pas l'élément `x` à l'arbre `a` s'il est déjà dedans. □

**Exercice 9** Écrire une fonction `compte(x, a)` qui renvoie le nombre d'occurrences de `x` dans l'ABR `a`. On ne suppose pas que l'arbre `a` a été construit à partir de la fonction `ajoute`, mais seulement qu'il s'agit d'un ABR. Cela veut dire qu'une valeur égale à la racine peut se trouver encore dans le sous-arbre gauche autant que dans le sous-arbre droit. Cela étant, on s'attachera à ne pas descendre dans les sous-arbres dans lesquels on est certain que la valeur `x` ne peut apparaître. □

**Exercice 10** Écrire une fonction `remplir(a, t)` qui ajoute tous les éléments de l'arbre `a` dans le tableau `t`, dans l'ordre infixe. Chaque élément `x` est ajouté au tableau `t` avec `t.append(x)`. Ajouter ensuite une méthode `lister(self)` à la classe `ABR` qui renvoie un nouveau tableau contenant tous les éléments de l'ABR `self` par ordre croissant. □

**Exercice 11** En utilisant l'exercice précédent, écrire une fonction `trier(t)` qui reçoit en argument un tableau d'entiers et renvoie un tableau trié contenant les mêmes éléments. Quelle est l'efficacité de ce tri ? □