

## Structures de données et algorithmes

### Feuille de TD n°3

#### Arbres binaires et récursion

Pour les besoins de cette feuille d'exercices, on considère que les arbres binaires sont implémentés à l'aide de la structure et du type suivants.

```
1 #include "item.h"
2
3 struct binary_tree {
4     item label; /* etiquette de la racine */
5     struct binary_tree *left; /* adresse du sous-arbre gauche (NULL s'il est vide) */
6     struct binary_tree *right; /* adresse du sous-arbre gauche (NULL s'il est vide) */
7 };
8 typedef struct binary_tree *link;
```

### Exercice 1. Arbres binaires : dénombrement, feuilles et hauteur

a) On suppose que `item` ne peut prendre qu'une seule valeur. Soit  $C_n$  le nombre d'arbres binaires (deux à deux distincts) de taille  $n$ , c'est-à-dire à  $n$  sommets. Pour  $n = 0, 1, 2, 3, 4$ , dessinez tous les arbres de taille  $n$  et déduisez-en la valeur de  $C_n$ .

b) Écrivez la relation de récurrence (complète) liant  $C_{n+1}$  à  $C_0, C_1, \dots, C_n$ , pour  $n \geq 0$ .

c) Une feuille est un sommet dont les sous-arbres gauche et droit sont vides. Calculez précisément le nombre minimal et le nombre maximal de feuilles d'un arbre binaire de taille  $n$ , pour  $n \geq 1$ .

d) La hauteur d'un arbre binaire non vide est la longueur maximale d'une branche, c'est-à-dire la longueur maximale d'un chemin menant de la racine à une feuille. Calculez précisément la hauteur minimale et la hauteur maximale d'un arbre binaire de taille  $n$ , pour  $n \geq 1$ .

### Exercice 2. Arbres binaires et algorithmes récursifs

a) Écrivez la définition d'une fonction C `size_binary_tree` qui reçoit en entrée l'adresse d'un arbre binaire et renvoie sa taille.

b) Écrivez la définition d'une fonction C `height_binary_tree` qui reçoit en entrée l'adresse d'un arbre binaire et renvoie sa hauteur. [N.B. Par convention, la hauteur d'un arbre binaire vide est  $-1$ ].

c) On suppose qu'il existe un ordre total sur les valeurs de `item` et que la macro `less(A, B)` est évaluée à « vrai » si l'item  $A$  est plus petit que l'item  $B$ . Écrivez la définition d'une fonction C `max_binary_tree` qui reçoit en entrée l'adresse d'un arbre binaire **non vide** et renvoie la valeur maximale de ses étiquettes.

d) Écrivez la définition d'une fonction C `count_leaves_binary_tree` qui reçoit en entrée l'adresse d'un arbre binaire et renvoie le nombre de ses feuilles. [N.B. Le nombre de feuilles d'un arbre binaire vide est 0].