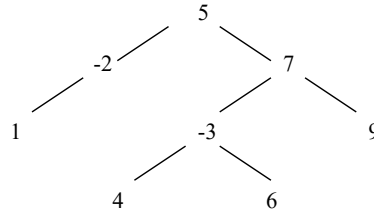


TDTP 7 : suite des arbres

Soit a l'arbre suivant :



À préparer avant la séance

- Définir une fonction qui multiplie par deux les valeurs des nœuds d'un arbre de nombres.

`(fois2 a) → (10 (-4 (2 () ())) ()) (14 (-6 (8 () ())) (12 () ())) (18 () ()))`

À faire pendant la séance

- TD** • Définir une fonction qui construit le miroir d'un arbre binaire : pour tous les nœuds de l'arbre, le fils gauche devient le fils droit et le fils droit devient le fils gauche.

`(miroir a) → (5 (7 (9 () ())) (-3 (6 () ())) (4 () ())) (-2 () (1 () ()))`

- TD** • Définir une fonction qui calcule la hauteur d'un arbre, définie comme la longueur du plus long chemin entre la racine et les feuilles. On pourra utiliser la définition récursive suivante : la hauteur d'un arbre est 1 + le maximum des hauteurs des sous-arbres (gauche, droit), la hauteur d'une feuille étant zéro.

`(hauteur a) → 3`

- TD** • Définir une fonction qui calcule le minimum d'un arbre binaire de nombres.

`(minimum a) → -3`

- TD** • Définir une fonction qui, étant donné un arbre de nombres, remplace la valeur de chaque nœud par la somme des feuilles accessibles depuis ce nœud.

`(remplace-feuilles a) → (20 (1 (1 () ())) ()) (19 (10 (4 () ())) (6 () ())) (9 () ()))`

- TD** • Définir une fonction qui, étant donné un arbre, remplace la valeur de chaque nœud par la profondeur de ce nœud (c'est-à-dire la longueur du chemin de la racine à ce nœud).

`(remplace-prof a) → (0 (1 (2 () ())) ()) (1 (2 (3 () ())) (3 () ())) (2 () ()))`

Pour s'entraîner (exercices supplémentaires facultatifs)

- Définir une fonction qui, étant donné un arbre de nombres, remplace les valeurs négatives par 0.

`(met0 a) → (5 (0 (1 () ())) ()) (7 (0 (4 () ())) (6 () ())) (9 () ()))`

- Définir une fonction qui, étant donné un arbre binaire de nombres, retourne une liste composée de cet arbre dont toutes les valeurs sont désormais positives et du nombre de changements qui ont été effectués pour obtenir ce nouvel arbre.

`(arbre-positif a) → ((5 (2 (1 () ())) ()) (7 (3 (4 () ())) (6 () ())) (9 () ())) 2)`

- Définir une fonction qui, étant donné un arbre d'entiers, renvoie un arbre dont la valeur de chaque nœud est une liste de deux nombres représentant le nombre d'éléments pairs et impairs présents dans les sous-arbres et lui-même.

`(pair-impair a) → ((3 5) ((1 1) ((0 1) () ())) ()) ((2 3) ((2 1) ((1 0) () ())) ((1 0) () ())) ((0 1) () ()))`