

Exercice B type CCINP

24 mars 2026

On considère dans cet exercice des arbres binaires tels que pour chaque noeud, le sous arbre gauche possède soit le même nombre d'éléments que le sous arbre droit soit exactement un de plus. On demandera de plus que pour chaque noeud, la valeur stockée soit inférieure ou égale à celles de ses fils. Un tel type d'arbre est appelé arbre de Braun.

On utilisera le type `type braun = E | N of (braun * int * braun)`

1. Déterminer les squelettes des arbres de Braun à deux noeuds. Même question pour trois et quatre noeuds.
2. Montrer qu'il n'y a qu'un seul arbre de Braun à n noeuds pour tout entier naturel n .

On admet que tout arbre de Braun non vide à n noeuds et de hauteur h vérifie : $h = \lfloor \log(n) \rfloor$.

3. Ecrire une fonction `hauteur : braun -> int` qui calcule la hauteur d'un arbre de Braun. On attend une complexité logarithmique en la taille de l'arbre.
4. Ecrire une fonction `get_min : braun -> 'a` qui renvoie l'élément de valeur minimale d'un arbre de Braun.
5. Une fonction `insert : int -> braun -> braun` qui ajoute un élément dans l'arbre de Braun et renvoie un arbre de Braun qui satisfait bien les propriétés de la définition qui vous est fournie. Prouver sa correction.
6. Ecrire une fonction `replace_min : int -> braun -> braun` qui prend en argument une valeur x et un arbre de Braun non vide t et renvoie un arbre de Braun contenant x et tous les éléments de t sauf le plus petit.
7. Ecrire une fonction `extract : braun -> int * braun` qui extrait un élément arbitraire de l'arbre de Braun tout en respectant la structure d'arbre de Braun pour le résultat. La fonction renvoie la valeur extraite et l'arbre privé de cette valeur qui doit être un arbre de Braun. (si on est gêné par cette question on pourra chercher à traiter la suivante qui permettra de comprendre où on veut en venir).
8. Ecrire une fonction `merge : braun -> braun -> braun` qui réalise la fusion de deux arbres de Braun l et r dont on sait que le nombre de noeuds de l est compris entre le nombre de noeuds de r et cette valeur plus un. La fonction devra renvoyer un arbre de Braun satisfaisant les propriétés.
9. Ecrire une fonction `extraire_min : braun -> braun` qui retire l'élément de valeur minimale d'un arbre de Braun.