

TP : structures de données d'arbre

Léo BATY

Exercice 1.

cf. `arbre.cpp` et `arbre.h`

Exercice 2.

La suite 12 8 4 9 23 17 15 correspond au parcours en profondeur d'abord en entrant.

Exercice 3.

Cas d'erreur pour les fonctions de **intTree** :

- `getSon` : la position n'existe pas ($pos < 0$ ou $pos > sons.size()$)
- `setSon` : pareil que pour `getSon`
- `removeLastSon` : si `sons` est vide, cela peut causer une erreur

3.2)

Pour `getSon` et `setSon` on peut signaler une erreur par exception `std::out_of_range`. Pour `removeLastSon` on peut lancer une exception `std::length_error`.

Exercice 4.

On ne peut pas séparer la classe template `Tree` en deux fichiers séparés `tree.h` et `tree.cpp`, car pour les classes templates, le code est régénéré pour chaque instance de e template. Il faut donc regrouper déclarations et définitions de la classe template `Tree` dans un seul fichier `tree.h`.

Exercice 5.

La profondeur maximale d'un arbre est égale au maximum des profondeurs maximales de chaque sous-arbre fils plus un. Ainsi, un programme récursif avec un parcours en profondeur en descendant convient.

La profondeur minimale d'un arbre est égale au minimum des profondeurs minimales de chaque sous-arbre fils plus un. Ainsi, un programme récursif avec un parcours en profondeur en descendant convient aussi.