

# Rapport Algo TP1

Louis Hémadou

October 22, 2019

## Exercice 2

La suite "12 8 4 9 23 17 15" correspond à un parcours en profondeur de l'arbre. En partant de la racine, on explore les fils de gauche à droite jusqu'à trouver un-cul-de sac. On revient alors jusqu'au dernier noeud exploré contenant plusieurs fils puis on explore les autres fils de la même manière.

## Exercice 3

3.1) Tout d'abord, il peut y avoir une erreur si on donne un paramètre du mauvais type au constructeur, à setData ou à setSon. Le compilateur sait néanmoins repérer ces erreurs efficacement.

Dans getSon et setSon, il faut bien prendre garde que  $0 < pos < nbSons()$ , ou le programme essaiera d'accéder à un élément inexistant.

Aussi, il est nécessaire de vérifier que le noeud possède bien un fils en appelant la fonction removeLastSon.

3.2) Dans le cas de getSon, il est possible de signaler l'erreur par valeur de retour, il serait cependant compliqué d'y ajouter un statut d'erreur (utilisation de std::pair). Dans le cas de setSon et removeLastSon, il serait assez simple d'ajouter un statut d'erreur, en retournant un booléen plutôt qu'un void.

3.3+3.4) Dans les deux cas, on peut et on va signaler les erreurs par exception, dans un souci de simplicité.

## Exercice 4

4.2) Dans le cas où on fait une classe template, il n'est pas possible de séparer la définition de la classe et la définition des méthodes dans deux fichiers différents.

4.3) Les erreurs rencontrées avec `Tree<T>` sont les mêmes que celles rencontrées avec `IntTree`. Le signalement des erreurs par exception étant suffisamment générique, il n'est pas nécessaire de modifier la gestion d'erreur.