

Notes sur TP9 à 12 PRG 2023/2024 (Arbres binaires)

Questions préliminaires à rendre AU DÉBUT de la 1ere séance de TP :

- un exemple d'intersection sur 2 arbres représentant des images + le code de la méthode *intersection* sont à montrer à votre enseignant de TP au début de la première séance.
- cette méthode n'a pas besoin d'être testée avant de la rendre : nous voulons tester votre compréhension du sujet et votre capacité à concevoir un algorithme pertinent, répondant à de multiples contraintes. Notez que ce sont aussi les conditions d'écriture de tels algorithmes en CC.
- le retour sur ces questions doit vous permettre d'identifier ce que vous n'avez pas compris dans notre problématique des listes (imposée, la même qu'en TD) et dans les multiples contraintes demandées. Attention, nous ne vous donnerons pas la solution, mais les indications sur vos incompréhensions seront pointées.

Réalisation du TP Arbres :

- 4 séances (d'1h30) de TP sont consacrées au TP Arbres binaires: semaines 42 et 43.
- dans un premier temps, vous devez réaliser des méthodes de la classe `Image.java` (résolution de problèmes sur les arbres binaires, dit au niveau abstrait) avec les méthodes arbres et les méthodes itérateurs d'arbres fournies par les enseignants (*BinaryTree.class* dans *TP-arbres.jar*).
- dans un second temps, vous devez réaliser vous-même l'implémentation des méthodes arbres et des méthodes itérateurs d'arbres (celles vues en PRG). Attention, ne mélangez pas les deux étapes : votre TP serait trop compliqué à mettre au point. Procéder par étapes, et incrémentalement, est essentiel dans le savoir faire d'un.e programmeur.se.
- le **TP Arbres est à rendre au plus tard le mardi 7/11/2023 à 20h sur Moodle.**

Archives disponibles :

Vous trouverez sur Moodle (L3Info-PRG) les fichiers concernant le TP Arbres. L'archive **tp_arbres.tar** vous permet :

- de définir un projet Java sous Eclipse (File → New → Java Project) et d'importer cette archive à la racine du projet (File → import → Archive File → `tp_arbres.tar`). Vous obtiendrez ainsi la structure prévue.
- d'obtenir
 - **TpArbres.java** : ne pas modifier, c'est un utilitaire de test « à la place » d'un utilisateur de nos fenêtres graphiques (la structure sous-jacente en arbres est transparente pour l'utilisateur). A utiliser absolument pour vous aider dans la phase de mise au point ! Cet

utilitaire permet de visualiser à la fois les images résultant d'opérations (*inverseVideo*, *intersection*, etc.) et les arbres correspondant.

- **Image.java** : méthodes à compléter, respecter la structure imposée qui permet le fonctionnement de TpArbres.java ET des tests unitaires.

Les utilitaires dont vous avez besoin sont réunies dans **tp_arbres.jar** : cela vous permet d'utiliser les méthodes des enseignants pour les méthodes arbres et pour les méthodes itérateurs (implémentées différemment de ce que l'on vous demande).

- ➔ Modifier votre classpath pour y avoir accès (à la racine du projet, Build Path → Configure Build Path → add external jar).
- ➔ **N.B.** La classe **Node.java** est également fournie pour que vous compreniez le fonctionnement de l'utilitaire Node.class.

Dans un second temps, un squelette BinaryTree.java vous sera fourni pour implémenter vos propres méthodes arbres et itérateur.

- ➔ Vous devrez alors **re-tester votre TP avec vos propres méthodes arbres et itérateurs.**

Lisez attentivement le sujet de TP pour ne pas oublier une consigne. Vous devez compléter les .java Image et BinaryTree et tester vos programmes .