INF2010 - Structures de données et algorithmes

Travail Pratique 5

Graphes

Département de génie informatique et logiciel
École Polytechnique de Montréal

Automne 2020

Objectifs

- Apprendre le fonctionnement d'un graphe
- Comprendre la complexité temporelle et spatiale d'un algorithme qui utilise des graphes
- Utiliser les concepts associés aux graphes dans des problèmes complexes

Pour ce laboratoire, il est recommandé d'utiliser l'IDE IntelliJ offert par JetBrains. Vous avez accès à la version complète (Ultimate) en tant qu'étudiant à Polytechnique Montréal. Il suffit de vous créer un compte étudiant en remplissant le formulaire au lien suivant:

https://www.jetbrains.com/shop/eform/students

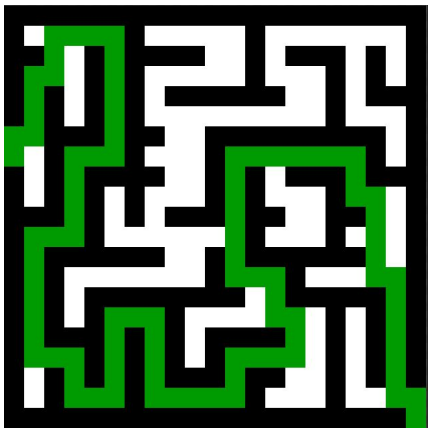
La correction du travail pratique sera partiellement réalisée par les tests unitaires implémentés dans les fichiers sources fournis. La qualité de votre code ainsi que la performance de celui-ci (complexité temporelle) seront toutes deux évaluées par le correcteur. Un barème de correction est fourni à la fin de ce *.pdf.

ATTENTION! ATTENTION! ATTENTION!
Pour ceux qui voudraient déposer leur laboratoire sur **GitHub**, assurez-vous que vos répertoires soient en mode **privé** afin d'éviter la copie et l'utilisation non autorisée de vos travaux. **Un répertoire public peut mener à une sanction de plagiat**.

Question d’entrevue 1 : Solveur de labyrinthe

Cet algorithme permet de trouver la longueur du chemin le plus court pour sortir d'un labyrinthe.

- **Carte de jeu** *maze* : Planche de jeu composée de carreau de labyrinthe
- **Carreau de labyrinthe** *Tile* : Sous-partie du labyrinthe représentant un morceau de plancher, un mur ou une entrée/sortie



Votre chemin doit commencer au point d'entrée et terminer au point de sortie . Les points d'entrée et sortie sont interchangeables. Votre chemin ne peut pas passer sur un carreau de labyrinthe qui est un mur.

Entrées

- Matrice de forme M x N (attribut *maze*) où chaque valeur représente une sous-partie (Plancher, mur ou entrée/sortie)

Sortie

Distance du chemin le plus court pour résoudre le labyrinthe

Pour bien implémenter l’algorithme, suivez les tests contenus dans MazeTest.java **dans l’ordre de leur définition**.

Question d’entrevue 2 : Détaille mon monde

Cet algorithme permet de regrouper des pays selon leur continent en sélectionnant un algorithme *breadth-first search* ou bien *depth-first search*.

- **Carte du monde** : Surface plane, plus précisément un rectangle, entourée d'eau
- **Continent** : Territoire entouré d'eau
- **Pays** : Sous-partie indissociée d'un continent

Chaque pays porte un nom distinct, soit un entier positif non-nul unique. Un pays ne peut faire partie que d'un seul continent.

Entrées

- Matrice de forme M x N (attribut *world*) où chaque valeur représente une région

  0 => Région d’eau

  Valeur positive non-null => Région du pays associé à cette valeur
- Booléen (attribut *isBreadthFirstSearch*) déterminant si l’algorithme à utiliser est le *breadth-first search* ou le *depth-first search* Sortie

La valeur de retour est une liste des continents ordonnés dans leur ordre d’apparition de gauche à droite, de haut en bas. Chaque continent est une liste de pays ordonnés selon leur valeur.

Pour bien implémenter l’algorithme, suivez les tests contenus dans DetailMyWorldTest.java **dans l’ordre de leur définition**.

Question d'entrevue 3 : k^{e} plus petit élément

Entrées

- Matrice de forme M x N (attribut *matrix*) où chaque valeur doit respecter les règles suivantes :
  - matrix[i][j] <= matrix[i + 1][j]
  - matrix[i][j] <= matrix[i][j + 1]
- Entier (attribut *k*) représentant la position de la valeur à retourner si la matrice serait mise dans un tableau 1D trié.

  Supposons k = 3

1	2	3
4	5	6
7	8	9

1	2	3	4	5	6	7	8	9

Sortie

Élément à la position *k* si la matrice était mise dans un tableau 1D trié.

En d'autres mots, on retourne le k^{e} plus petit élément de *matrix*.

Contraintes

Supposons une matrice (*matrix*) de forme quelconque M x N.

- Complexité temporelle (temps) : O(k log max(m, n))
- Complexité spatiale (mémoire) : O(log max(m, n))

Expliquez la complexité de votre algorithme dans l'en-tête de la fonction *findKthSmallestElement*.

Pour bien implémenter l'algorithme, suivez les tests contenus dans KthSmallestElementTest.java **dans l'ordre de leur définition**.

Ce que vous pouvez faire..

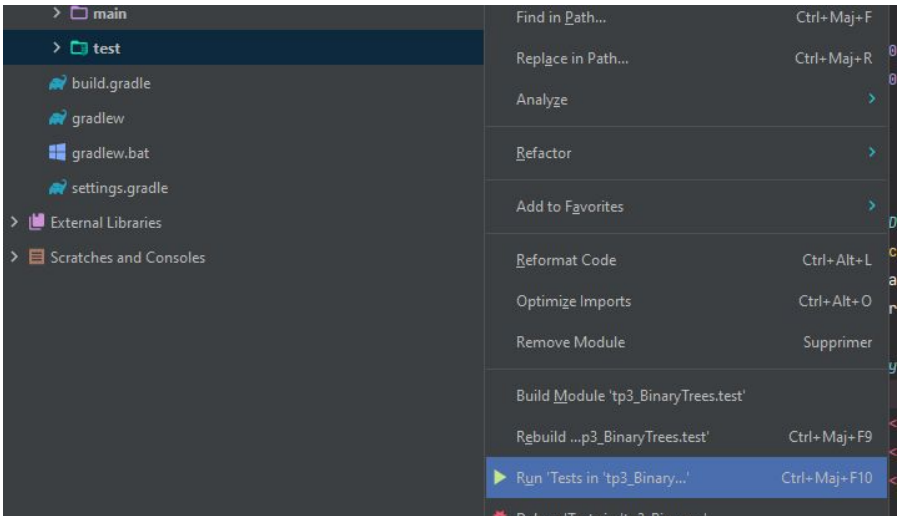
Dans ce laboratoire, il sera particulièrement important d'avoir un code bien auto documenté en encapsulant certains comportements dans des fonctions ou même des classes. **La seule restriction est de ne pas modifier l'API public**. Toutes autres modifications sont autorisées.

Par exemple :

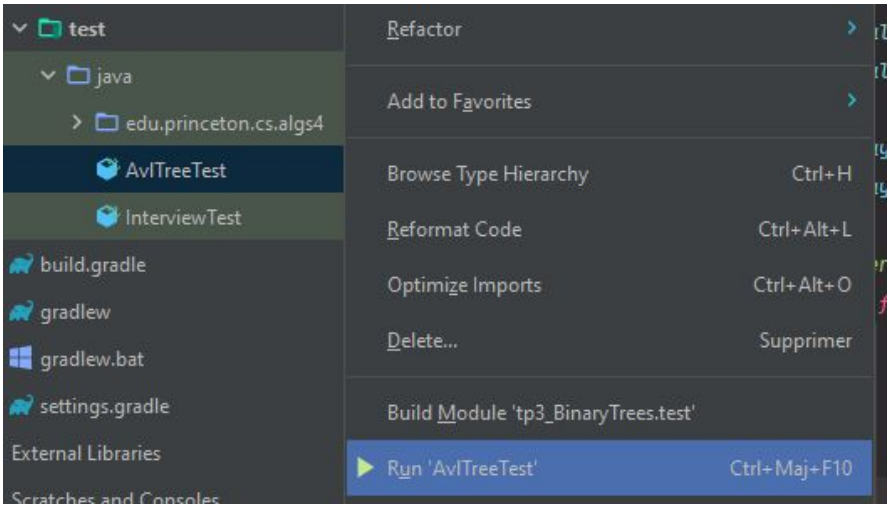
- Ajouter une fonction privée
- Ajouter un attribut privé
- Créer une classe
- Créer un enum
- …

Si vous décidez de créer de nouveaux fichiers, mettez-les dans le dossier associé à la question qui les utilise.

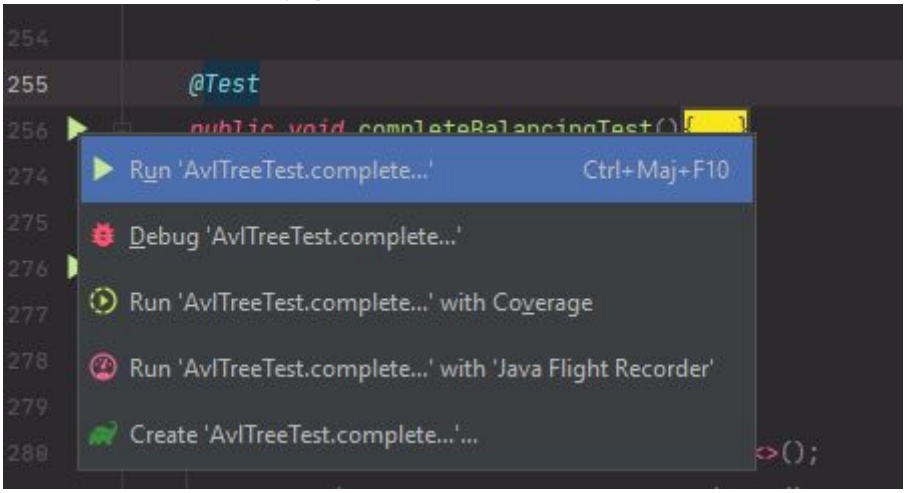
Pour créer une *build configuration* contenant tous les tests



Pour créer une *build configuration* contenant un tester



Pour créer une *build configuration* contenant un test



Barème de correction

Solveur de labyrinthe	Tests	/4
Détaille mon monde	Tests	/5
k^e plus petit élément	Tests	/4
	Complexité et explication	/6
Qualité du code		/1
		/20

Un chargé s'assurera que votre code ne contourne pas les tests avant de vous attribuer vos points. La note de la catégorie « Tests » est proportionnelle au ratio $\frac{Nombre\ de\ tests\ réussis}{Nombre\ de\ tests}$.

Qu'est-ce que du code de qualité ?

- Absence de code dédoublé
- Absence de *warnings* à la compilation
- Absence de code mort
- Respecte les mêmes conventions de codage dans tout le projet
- Variables, fonctions et classes avec des noms qui expliquent leur intention et non leur comportement

Instructions pour la remise

Veuillez envoyer un *.zip de votre dossier main.

Vos fichiers devront être compressés dans une archive *.zip. Le nom de votre archive devra respecter la formule suivante où MatriculeX < MatriculeY :

inf2010_lab5_MatriculeX_MatriculeY

Chaque jour de retard créera une pénalité additionnelle de 20%. Aucun travail ne sera accepté après 4 jours de retard.