

IQ02E020 Exploration algorithmique d'un problème

SAE 2.02 2023 (Graphes)

Recettes

Contact : mikal.ziane@gmail.com (les adresses officielles sont en panne)

Attention : le non-respect des dates limites indiquées ci-dessous entraînera des points de pénalités sur la note.

Quand et Qui ?

Les recettes auront lieu lors de la dernière séance de **TP** : le projet devra donc être rendu sur GitHub jusqu'au **lundi 22 mai matin 8h**. Même si tous les membres de l'équipe ne sont pas présent la soutenance aura lieu si au moins un des membres est présent. **La note de 0 sera attribuée aux absents qui n'ont pas de justification valable**. Sauf cas exceptionnel, la note sera la même pour tous les membres de l'équipe (mais l'exercice du DST sera évidemment individuel).

Sur quelle machine ?

Les recettes seront a priori exécutées sur un ordinateur portable de l'équipe, pensez à en apporter un, sauf si les machines de l'IUT sont effectivement utilisables (ce qui nécessite des tests avant la recette et non pas au dernier moment). En cas d'impossibilité prévenez votre chargé de TP pour voir s'il peut se procurer un pc.

Il doit être possible de faire la recette à partir de la version qui a été mise sur GitHub au plus tard à la date limite indiquée plus haut.

Que rendre ?

Contrairement à ce qui était prévu au tout début de la SAE il n'y a pas d'étude de performance sur des ajouts de nœuds. L'étude de performance porte uniquement sur Dijkstra sur les 4 implémentations des graphes et sur les 3 différentes sortes de graphes fournis (DorogovtsevMendes, Barabasi-Albert, full-connected). Ce qui change principalement entre ces types de graphes, c'est le nombre moyen d'arcs par nœud.

Tout ce qui est à rendre doit être sur GitHub :

- un fichier **readme.md** avec les informations suivantes :
 - Le nom et prénom des membres de l'équipe. Ils ne doivent pas avoir changé depuis le début du projet (sauf accord du chargé de TP)
 - Quelles classes de graphes parmi les 4 demandées ont été codées ?
 - Est-ce que Dijkstra passe les tests fournis sur Moodle avec tous les graphes fournis ? jusqu'à quelle taille (selon le type¹ de graphe).
- Un pdf avec le diagramme d'architecture (paquetages et classes avec les associations entre les classes y compris les dépendances, **sans** les méthodes et les attributs). Il est conseillé de reprendre le découpage en paquetage de la dernière version des sources fournies sur Moodle le 14 mai.
- Le code compilable avec suffisamment de tests unitaires (au moins ceux fournis sur Moodle).

¹ Les types de graphes fournis sur Moodle sont DorogovtsevMendes (répertoire orig), Barabasi, et full-connected en plus des graphes dans le répertoire autres qui servent à tester certains bugs de Dijkstra.

- Les **tests unitaires** et le **main** fournis sur Moodle doivent tourner sur les graphes fournis aussi sur Moodle. **Ce sont eux qui seront en principalement utilisés pour tester votre projet lors de la recette.** En commentant un ou plusieurs des « prototypes » dans la fonction main on doit pouvoir tester indépendamment chaque type de graphe si on le souhaite. Pour tester tel ou tel graphe il doit être possible, comme sur le code fournit sur Moodle, de simplement copier des graphes et des réponses d'un sous répertoire en le plaçant à la racine c'est-à-dire en les remontant dans les répertoires « graphes » et « reponses ».
- Un fichier **performances.pdf** similaire au fichier perfs.pdf fournit sur Moodle (inutile de remettre les illustrations).

Critères de notation

Rappel : la note de la SAE combine la note globale de maths et la note globale d'informatique. La note d'informatique est à 50% la note qui sera donnée suite à la recette et à 50% la note de l'exercice graphes du DST. Cet exercice comptera à la fois pour la SAE et pour le DST et suppose que vous connaissez parfaitement tout ce qui a été fourni sur Moodle, notamment tout le code et les tests fournis.

Est-ce que tout ce qui a été demandé dans ce document a été rendu en temps et en heure sur GitHub ?

Le **diagramme d'architecture** est-il complet avec notamment les **dépendances** ?

Le rapport **performances.pdf** est-il suffisamment fourni ?

Est-ce que tous les **tests** fournis sur Moodle, ainsi que la fonction **main**, tournent correctement ?

Est-ce que les 4 types de graphes sont codés ?

Est-ce que Dijkstra est **générique** et ne dépend donc pas de chaque type de graphe mais uniquement de l'interface IGrapheConst ?

Est-ce que l'**encapsulation** des données est respectée ?

Est-ce que le **confinement**² des données est respecté ?

Est-ce que le code respecte les **consignes habituelles** sur la non-duplication du code, les nombres magiques, la taille maximum des méthodes (10 à 15 lignes sauf raison particulière), le nommage clair des entités etc. ?

² Confiner les données d'un objet c'est ne pas retourner de pointeurs dessus. On peut retourner une copie ou à la limite un pointeur sur une donnée non mutable.