R2.07 Graphes

TP2 – Familles de graphes

2024-2025

Au démarrage

Vous avez utiliser l'outil maven lors du dernier tp. En particulier, vous avez pu expérimenter les commandes `mvn compile` (pour compiler), `mvn exec:java` pour exécuter une classe spécifique et `mvn test` pour lancer vos tests. Cependant, nous vous avions faciliter la vie en vous fournissant un projet déjà préparé.

Aujourd'hui, c'est vous qui allez mettre en place votre projet. Pour cela, maven vous permet d'utiliser des **archetypes** qui génèrent une arborescence ainsi qu'un fichier de configuration `pom.xml` dans lequel sera renseigné toutes les dépendances de votre projet. n

Dans notre cas, nous allons partir d'un archetype pour JGrapht. Voici la commande à utiliser :

```
mvn archetype:generate -DgroupId=fr.univ_orleans.iut45.r207.tp2 \
   -DartifactId=tp2 \
   -DarchetypeGroupId=org.jgrapht.archetypes \
   -DarchetypeArtifactId=maven-archetype-jgrapht \
   -DarchetypeVersion=1.2.0
```

Vous devriez obtenir l'arborescence suivante :

```
tp2/
├── pom.xml
└── src
    ├── main
    │   └── java
    │       └── fr
    │           └── univ_orleans
    │               └── iut45
    │                   └── r207
    │                       └── tp2
    │                           └── App.java
    └── test
```

```
└── java
    └── fr
        └── univ_orleans
            └── iut45
                └── r207
                    └── tp2
                        └── AppTest.java

16 directories, 3 files
```

Cependant, la version de JGrapht est un peu ancienne. De plus, seul la dépendance au package `core` est ajouté par défaut. Vous pouvez récupérer sur Celene une version du `pom.xml` plus adaptée à ce tp.

Enfin, pensez à faire vos commit et vos tests!

Voisinages

1. Déterminer l'union des voisinages de deux sommets u et v. Si l'un des deux sommets n'existe pas, alors on renverra un ensemble vide.
2. Déterminer les voisins de u qui n'appartiennent pas à v.
3. Deux sommets sont jumeaux s'ils ont le même voisinage. Ecrire une fonction qui permet de déterminer si deux sommets sont jumeaux.

Sous-graphes

1. A partir d'un graphe G et d'une liste de sommets S, renvoyer le sous-graphe induit par S
2. A partir d'un graphe G d'une liste d'arêtes S, renvoyer $G \setminus S$
3. Soit C une chaine. Ecrire une fonction qui renvoie vrai si la chaine est élémentaire.
4. Soit C une chaine. Ecrire une fonction qui extrait et renvoie une chaine élémentaire à partir de C

Applications

Reprenez le graphe des héros marvel du tp précédent et testez les fonctions précédentes!

Pour les plus rapides

Génération de familles de graphes

1. Générer un cycle à n sommets

2. Générer un graphe complet à n sommets
3. Générer un graphe bi-partie aléatoire (soit $u \in A$ et $v \in B$, alors $\{u, v\} \in E$ avec proba p)

Graphes orientés et graphes pondérés

Vous utiliserez les classes suivantes :

```
SimpleDirectedGraph
SimpleWeightedGraph
SimpleDirectedWeightedGraph
```

pour instancier respectivement des graphes orientés (non pondérés), des graphes pondérés (non orientés) et des graphes pondérés orientés.

Écrivez ensuite une méthode pour chacune des questions suivantes.

1. Soit $W = (G, w)$ un graphe pondéré et P une suite de sommets. Renvoyer la somme des poids des arêtes du chemin induit par P.
2. Soit D un graphe orienté pondéré et P une suite de sommets. Déterminer si ces sommets induisent un chemin dirigé. Si c'est le cas, renvoyer la sommet des poids des arcs.