

IR2 - Algorithmique des graphes

Fiche 1 - Représentation des graphes

L'objectif de ce TP est d'implanter différentes structures pour représenter des graphes ainsi que les opérations de base pour les manipuler afin de construire un cadre pour concevoir des algorithmes dans les prochaines séances. Le langage de programmation est le Java.

Les graphes considérés dans ce TP sont étiquetés par des entiers allant de 0 à $n - 1$, n étant le nombre de sommets du graphe. En revanche, ils peuvent être orientés ou non, et pondérés ou non.

Les opérations de base de manipulation de graphes à implanter dans ce TP sont :

1. la création d'un graphe (resp. graphe orienté, graphe pondéré) à n sommets, sans arête (resp. arc) ;
2. l'ajout d'une arête (resp. d'un arc, arête pondérée) à un graphe (resp. graphe orienté, graphe pondéré) ;
3. la suppression d'une arête (resp. d'un arc, d'une arête pondérée) à un graphe (resp. graphe orienté, graphe pondéré) ;
4. le test d'appartenance d'une arête (resp. d'un arc, d'une arête pondérée) à un graphe (resp. graphe orienté, graphe pondéré) ;
5. l'affichage d'un graphe (resp. graphe orienté, graphe pondéré), en représentation sagittale. Pour ce faire, il est possible de redéfinir la méthode `toString` de manière à ce qu'elle produise une chaîne de caractères du type

```
digraph G {
    1 -> 2 [label="1"];
    2 -> 3 [label="3"];
    2 -> 4 [label="6"];
    3 -> 1 [label="1"];
    4 -> 2 [label="2"];
};
```

(remplacer `->` par `--` dans le cas non orienté et supprimer `[label="*"]` dans le cas non pondéré) pour utiliser le programme `dot` qui permet de produire un affichage d'un graphe au format `.png`. Il suffit en effet d'écrire le résultat de `toString` dans un fichier suffixé par `.dot` puis de lancer la commande `dot FICHIER.dot -Tpng -o FICHIER.png` ;

6. le chargement d'un graphe à partir d'un fichier au format `dot`. Cette opération va être utile pour réaliser des tests.

Exercice 1. Représentation matricielle

Implanter le nécessaire pour représenter par matrice d'adjacence les graphes orientés.

Exercice 2. Représentation liste d'adjacence

Implanter le nécessaire pour représenter par liste d'adjacence les graphes, les graphes orientés et les graphes pondérés.