

Comme je l'ai écrit il y a quelques jours, le mini-projet porte sur la programmation de l'algorithme du parcours en profondeur (conformément aux feuilles numéro 6 et 7) afin de détecter les circuits dans un graphe orienté. **Ce projet est à rendre pour le 16 janvier 2022, délai de rigueur.**

### Travail à faire et contraintes de programmation

Quelques indications sur ce qui est demandé et certaines contraintes à respecter :

- les graphes que l'on manipule sont des graphes orientés, linéaires (sans arcs parallèles) et sans boucles ils seront représentés par listes d'adjacence (plus précisément, les listes de successeurs des sommets) ;
- les sommets d'un graphe d'ordre  $n$  seront les entiers de 1 à  $n$  (et pas de 0 à  $n - 1$ , attention !) ;
- un programme devra lire le graphe sur son entrée standard ou dans un fichier, dans un format habituel **strict** redonné plus bas, puis en utilisant un parcours en profondeur, détecter si ce graphe possède (au moins) un circuit et, dans ce cas, produire ce circuit par la liste, dans l'ordre du circuit, de ses sommets :  $x_1, x_2, x_3, \dots, x_k$  (si le circuit est de longueur  $k$ ), l'arc  $(x_k, x_1)$  refermant ce circuit.

Le format d'entrée des graphes est une suite de nombres entiers constituée de la manière suivante :

- un premier entier,  $n$ , qui est l'ordre du graphe considéré ;
- puis  $n$  suites de nombres entiers structurées ainsi :
  - un entier  $x$  compris entre 1 et  $n$ , représentant un sommet,
  - une suite d'entiers compris entre 1 et  $n$ , représentant la suite de successeurs du sommet  $x$ ,
  - l'entier 0 indiquant la fin de la suite de successeurs en question.

Vous avez des exemples de graphes décrits dans un tel format en examinant les fichiers d'exemples, donnés sur la page du cours, dans les fichiers d'extension `.alists`.

Tout autre format sera rejeté immédiatement par le programme<sup>1</sup> ; l'intérêt d'un tel format est que les graphes exemples peuvent être utilisés par tous les programmes sans avoir besoin de se préoccuper de telle ou telle conversion de format (une petite collection de graphes a déjà été déposée sur l'ENT, et vous en inventerez vous-mêmes d'autres).

Vous devrez également avoir programmé la génération aléatoire d'un graphe orienté telle qu'elle est décrite dans la feuille numéro 7. (En bref, vous devez avoir programmé toute la feuille 7.)

La programmation devra être soignée, chaque entité devant être soigneusement spécifiée et décrite (dans un document annexé au compte-rendu que j'appelle « rapport » dans la suite). Le projet doit être écrit en Java, si possible dans une version JDK-8. **Vous n'êtes pas autorisés à utiliser des composants tout faits, qui ne sont pas de votre cru** (sauf autorisation préalable).

**Le travail est à faire en binôme de préférence, prévenez-nous si vous le faites seul ;** lorsque vous rendez le travail, un seul rendu est exigé pour les deux membres du binôme. Aucun travail fait à trois ne sera accepté.

### Travail à rendre

Vous rendrez une archive zip ou tar ou tar.gz, au choix, à l'exclusion de tout autre format (des points seront déduits si l'archive n'est pas dans l'un de ces formats). Cette archive devra se décompresser pour créer un unique répertoire (portant les noms de famille en minuscules des deux membres du binôme, séparés par un tiret, et dans l'ordre alphabétique [par exemple, **naudin-pecquet**]) dans lequel se trouvera tout votre travail sous une forme immédiatement testable par nous. L'archive aura le même nom que le répertoire principal, avec l'extension idoine.

Dans ce répertoire principal, il devra y avoir, a minima, la racine de la hiérarchie de fichiers sources que nous pourrons compiler, accompagnée d'un makefile ou d'un shell-script permettant

1. Et par le correcteur !

de construire l'application ou bien d'une petite description de la commande à exécuter pour la compilation, si elle n'est pas trop complexe ; nous n'utiliserons aucun exécutable tout fait, il faut donc que nous puissions compiler sur nos machines (Java en version 8 maximum). Si vous utilisez un IDE, vous prendrez soin de nettoyer l'archive que vous nous envoyez de tous les fichiers spécifiques à votre IDE.

Le rapport de projet peut également apparaître dans ce répertoire principal, joint à l'archive des sources Java, il devra être fourni en format PDF (ou texte brut) à l'exclusion de tout autre format plus ou moins exotique (dans le cas contraire, nous considérerons qu'il n'y a pas de rapport).