

Algorithmes et structures de données 2

Laboratoire n°3 : Graphes orientés

07.11.2014

Introduction

Pour ce laboratoire, vous devrez implémenter une application, se basant sur les principes de graphes orientés, de cycles orientés et de tri topologique, permettant de résoudre un problème d'ordonnement de cours en fonction de leur prérequis.

Objectifs

- Implémentation d'une classe de détection de cycle ;
- Implémentation du tri topologique ;
- Utilisation sur des données réelles, un choix de cours de votre plan d'étude avec leurs prérequis afin de vérifier s'il est possible. Nous vous fournissons les fichiers *prerequis.txt* et *prerequis2.txt* contenant une sélection de cours.

Durée

- 4 périodes : A rendre le dimanche **23.11.2014** à minuit au plus tard.

Donnée

- Vous trouverez les structures et données fournies dans le dossier du cours :
<\\eistore1\profs\OCE\ ASD2 - 2014\Labos\Labo 3>
- Nous vous fournissons la classe *DiGraph* implémentant un graphe orienté par liste d'adjacence. Jetez-y un œil.
- Vous devrez réutiliser la structure *SymbolGraph* du laboratoire précédent, vous devrez adapter le constructeur afin de pouvoir parser les fichiers *prerequis.txt* et *prerequis2.txt*. Idéalement il faudrait un parseur générique, auquel il est possible de spécifier le séparateur.

- Nous vous fournissons l'interface de *DirectedCycle*, veuillez l'implémenter en gardant l'interface intacte. Cette classe permettra de détecter les cycles dans un graphe.
- L'interface *TopologicalSort* doit aussi être implémentée par vos soins. Vous devrez certainement utiliser une des structures rencontrées au précédent laboratoire pour réaliser cette classe.
- Vous pouvez utiliser toutes les structures de la librairie¹ std de c++, pour `-std=c++0x` ainsi que toutes les structures rencontrées lors des précédents laboratoires. N'hésitez pas non plus à re-parcourir une fois les slides du cours « Graphes Orientés ».
- Vous implémenterez la méthode *main()* du fichier *main.cpp*, celle-ci devra avoir une sortie équivalente aux exemples de la Fig. 1. La méthode *checkOrder()* fournie vous permet de vérifier votre solution par rapport au fichier d'entrée.

```
prerequis.txt est un DAG
Ordre topologique:
<cours_1> <cours_2> <cours_3> ...
Verification reussie
```

```
prerequis2.txt n'est pas un DAG
Cycle trouve:
<cours_1> <cours_2> <cours_3> ... <cours_1>
```

Figure 1 - Sorties de l'application

Rendu/Evaluation

Il n'y a pas de rapport à rendre pour ce laboratoire. Vous devrez par contre apporter une attention particulière aux commentaires dans votre code, ceux-ci devront permettre d'identifier les étapes principales de vos implémentations ainsi que les choix que vous avez effectués.

Adresse E-Mail de l'assistant : fabien.dutoit@heig-vd.ch

Bonne chance !

¹ <http://www.cplusplus.com/reference/stl/>