Algorithmes et structures de données 2
Laboratoire n°2 : Graphes orientés

10.10.2017

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'ordonnancement de modules 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 modules de cours de votre plan d'étude (IL) avec leurs prérequis afin de vérifier s'il est possible. Nous vous fournissons les fichiers *prerequis.txt* et *prerequis2.txt* contenant une sélection de modules.

Durée

- 4 périodes : à rendre sur le Moodle du cours au plus tard le dimanche **29.10.2017** à **23h55**.

Donnée

- Vous trouverez les structures et exemples fournis sur la page Moodle du cours : https://cyberlearn.hes-so.ch/course/view.php?id=10182

- 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 une implémentation dans le cas où vous n'auriez pas réussi à réaliser

cette partie. Si vous reprenez votre implémentation, veuillez indiquer que les méthodes *contains*, *index*, *name* et *adjacent* ne modifient pas les données en déclarant ces méthodes comme `const`.

- 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 pouvez utiliser toutes les structures de la librairie[1] std de c++, pour `-std=c++11` 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:
<mod_1> < mod_2> < mod_3> …
Vérification réussie
```

```
prerequis2.txt n'est pas un DAG
Cycle trouve:
< mod_a> < mod_b> < mod_c> … <mod_a>
```

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.

Adresses E-Mail des assistants :

- ASD2-1-A-L1 fabien.dutoit@heig-vd.ch
- ASD2-1-B-L1 michael.sandoz@heig-vd.ch
- ASD2-1-C-L1 raphael.racine@heig-vd.ch

Bonne chance !

[1] http://www.cplusplus.com/reference/stl/