

# LOG 2810

## STRUCTURES DISCRETES

### TP : GRAPHERS ET AUTOMATES

Session	HIVER 2021
Pondération	20 % de la note finale
Taille des équipes	3 étudiants
Date de remise du projet	20 avril 2021 (23h55 au plus tard)
Directives particulières	Soumission du livrable par moodle uniquement ( <a href="https://moodle.polymtl.ca">https://moodle.polymtl.ca</a> ).
	Toute soumission du livrable en retard est pénalisée à raison de 10% par jour de retard.
	Programmation C++
Les questions sont les bienvenues et peuvent être envoyées à : Abdelkader Zobiri ( <a href="mailto:abdelkader.zobiri@polymtl.ca">abdelkader.zobiri@polymtl.ca</a> ), Mohameth Ndiaye ( <a href="mailto:mohameth-alassane.ndiaye@polymtl.ca">mohameth-alassane.ndiaye@polymtl.ca</a> ).	

## 1 Connaissances requises

- Notions d'algorithmique et de programmation C++.
- Notions de théorie des graphes.
- Notions sur les automates.
- Notions sur les structures de données.

## 2 Objectif

L'objectif de ce travail pratique est de vous permettre d'appliquer les notions théoriques sur les graphes et les automates que nous avons vues en cours, sur des cas concrets mais hypothétiques, tirés de votre quotidien. A cet effet, il est question dans ce travail de :

- a) Permettre à une personne de recevoir une notification d'exposition Alerte COVID. Ce premier aspect est abordé dans la partie 5.
- b) Développer une application instructive pour enfant contenant un éditeur de texte 7.

## 3 Mise en situation

a) Avec la propagation du corona virus 2019, il est pertinent que les ingénieurs développent des applications de traçage pour mieux monitorer la propagation du virus. Or, les applications de traçage, dans leur fondement, sont tout simplement des graphes dont les sommets représentent des

personnes et dont les arcs représentent la puissance d'un signal Bluetooth qui estime la distance entre des personnes. Savoir manipuler les graphes de connexions serait donc un atout non seulement en entreprise, mais aussi en environnement de recherche, puisque c'est un domaine aux multiples applications. L'application Alerte COVID, nous servira de moyen d'apprentissage pour la manipulation des graphes. Il vous est donc demandé d'implémenter une application Alerte COVID.

b) L'ensemble des mots d'un langage forme un lexique. Dans les applications de suggestion automatique, le lexique peut être organisé sous la forme d'un automate à états finis, où chaque arc correspond à une lettre. L'objectif est ici de construire un automate à états finis capable de représenter le langage du lexique sur la pandémie de COVID donné en entrée, puis d'utiliser cet automate pour faire la suggestion de mots liés au COVID à l'utilisateur d'une application offrant un service de suggestion automatique de mots. Il s'agit donc de simuler l'une des fonctionnalités d'un correcteur automatique en utilisant l'automate correspondant au lexique. Vous avez été sollicité pour travailler sur le développement d'une application instructive pour enfant contenant un éditeur de texte.

#### Notes

- Les deux parties a) et b) seront intégrées ensemble dans une future tâche.

## 4 Description

### **Alerte COVID - Partie 1**

L'application Alerte COVID que vous devrez implémenter diffère légèrement de la version originale. Voici les contraintes de l'application :

- Un fichier individus.txt recense tous les individus que votre application doit prendre en compte, ainsi qu'un tag pour indiquer si un individu a eu un diagnostic (1) ou non (0). Ces deux données sont séparées par une virgule. On considère que tous les individus portent un nom unique qui les distingue.
- Un fichier contacts.txt recense tous les contacts existants entre les individus, ainsi que la distance entre des personnes qui varient entre 0 et 40 mètres.
- Le format de ces fichiers est détaillé dans l'annexe.
- L'application doit, permettre à une personne de vérifier si elle a été exposée au COVID au cours des 14 derniers jours. L'interaction se fait au moyen d'une interface console.
- Le contact entre deux personnes doit avoir été étroit, soit à moins de deux mètres.
- Un exemple d'affichage de réponse de l'application se trouve en annexe.

### **Jeu instructif - Partie 2**

- L'automate est utilisé pour énumérer toutes les formes possibles d'un mot donné.
- Il vous a été assigné comme travail, de livrer une interface console avec quelques fonctionnalités.
- La console doit permettre à l'enfant de saisir du texte (à partir du clavier). Au fur et à mesure que ce dernier écrit chaque lettre (sauf le séparateur "espace" et les signes de ponctuation), votre application doit lui afficher, sur la base du lexique (automate), toutes les formes possibles du mot déjà constitué. Par exemple, sur la base du lexique dont l'extrait est présenté dans la table 1, dès que l'utilisateur aura écrit la lettre "c", tous les mots figurant dans l'extrait doivent lui être affichés. Dès qu'il aura écrit successivement les lettres cas "c", "o", "n", les mots "confiné", "confirmé", "contact" doivent s'afficher à l'écran.

Mots
capacité
cas
cellule
clinique
confiné
confirmé
contact
viral

Table 1 : Extrait d'un lexique

- Il vous a aussi été demandé de tenir des métadonnées sur les mots. Il s'agit de prendre les précautions permettant de connaître la fréquence d'utilisation de chaque mot, ainsi que les dix mots les plus récemment utilisés par l'enfant. Ces données sont consultées par un parent sur demande.

## 5 Alerte COVID - partie 1

- C1. Écrire une fonction « `creerGrapheExposition()` » qui lit les fichiers texte contenant les informations sur les individus et les contacts physiques qu'ils ont eu, et génère le graphe d'exposition correspondant. Cette fonction prend en paramètre deux noms de fichiers à lire.
- C2. Écrire une fonction « `afficherGrapheExposition()` » qui affiche le graphe d'exposition selon le format présenté en annexe.
- C3. Écrire une fonction « `IdentifierExposition()` » qui permet de déterminer, en vous inspirant de l'algorithme de Dijkstra, si une personne X a été exposée au COVID par une autre Y. L'origine (X) et la destination (Y) sont passés en paramètres.
- C4. Écrire une fonction « `NotifierExposition()` » qui permet de déterminer si une personne a été exposée au COVID. Cette dernière est passée en paramètre.
- C5. Faire une interface qui affiche le menu suivant :
  - (a) Créer le graphe d'exposition.
  - (b) Afficher le graphe d'exposition.
  - (c) Afficher notification COVID
  - (d) Quitter.

### Notes

- Le programme doit toujours réafficher le menu, tant que l'option (d), ou « Quitter », n'a pas été choisie.
- L'utilisateur doit entrer un index valide, sinon le programme le signale et affiche le menu de nouveau.
- L'option (a) permet de lire de nouveaux fichiers d'individus et de contacts. Les noms des fichiers doivent être demandés à l'utilisateur. Le programme prend les noms des fichiers à lire à partir de la console, car l'utilisateur du programme peut donner les fichiers de son choix, en autant qu'ils respectent les formats en annexe.
- L'option (b) permet d'afficher dans la console le résultat de la lecture des fichiers et de la création du graphe d'exposition. Cette option ne peut pas être choisie avant l'option (a).
- L'option (c) permet de déterminer si une personne a été exposée au COVID. Pour ce faire, le nom d'une personne doit être demandé par la console. Cette option ne peut pas être choisie avant l'option (a).

## 6 Jeu instructif - partie 2

- C6. Créer un automate à partir d'un lexique. Le lexique est donné sous la forme d'un fichier .txt.
- C7. Créer une interface console qui permet à l'enfant de saisir du texte. Au fur et à mesure que l'enfant écrit un mot, votre application doit lui afficher, sur la base du lexique (automate), toutes les formes possibles du mot. Par exemple, sur la base du lexique dont l'extrait est présenté ci-haut, dès que l'enfant écrit la lettres "c", tous les mots figurant dans l'extrait doivent lui être affichés.
- C8. Proposer puis implémenter une façon d'associer à chaque mot deux labels. Le premier est un entier qui indique le nombre de fois qu'il a été utilisé. Le second est un entier dont la valeur est soit 0, soit 1. L'entier 0 indique que le mot ne fait pas partie des dix mots récemment utilisés. L'entier 1 indique que le mot fait pas partie des dix mots récemment utilisés. Les labels sont mis à jour dès que l'automate reconnaît le mot saisi par l'enfant.
- C9. Afficher sur demande, pour chaque mot du lexique, ses labels correspondant.
- (d) Créer automate.
  - (e) Saisir mot.
  - (f) Afficher statistiques.

### Notes

- L'option (d) permet de lire un nouveau lexique afin de créer l'automate correspondant. Il est possible que de nouveaux mots s'ajoutent ou s'enlèvent à la banque de lexiques, alors le mécanisme conséquent doit être disponible dans votre code. Pour lire un nouveau fichier de lexiques, le nom du fichier doit être demandé à l'utilisateur.
- L'option (e) permet à l'enfant de saisir des mots.
- L'option (f) permet d'afficher les statistiques sur l'état du système.
- Mettez l'option "Quitter" comme dernière option.

## 7 Main

La fonction principale de votre programme doit contenir un petit menu dirigeant l'utilisateur vers le numéro sur l'alert COVID ou le numéro sur le jeu instructif.

Faire une interface qui affiche le menu suivant :

- (a) Alerte COVID.
- (b) Jeu Instructif.
- (c) Quitter.

Les options « Quitter » du numéro sur l'alerte COVID et du numéro sur le jeu instructif ramènent vers ce menu. L'option « Quitter » de cette fonction principale termine l'exécution de votre programme.

### Notes

- Le programme doit toujours réafficher le menu, tant que l'option (c), ou « Quitter », n'a pas été choisie.
- L'utilisateur doit entrer un index valide, sinon le programme le signale et affiche le menu de nouveau.
- L'option (a) redirige l'utilisateur vers l'interface console du numéro alerte COVID.
- L'option (b) redirige l'utilisateur vers l'interface console du numéro du jeu instructif.

## 8 Livrable

Le livrable attendu est constitué du code source et du rapport de laboratoire. Le livrable est une archive (ZIP ou RAR ou tar.gz) dont le nom est formé des numéros de matricule des membres de l'équipe et groupe, séparés par un trait de soulignement (\_). L'archive contiendra les fichiers suivants :

- les fichiers .cpp ;
- les fichiers .h ;
- le rapport au format PDF ;

L'archive ne doit pas contenir de programme exécutable, de fichier de projet ou solution de Visual Studio, de répertoire Debug ou Release, etc. Les fichiers .cpp et .h suffiront pour l'évaluation du travail.

### 8.1 Rapport

Un rapport de laboratoire rédigé avec soin est requis à la soumission (format .pdf, maximum 8 pages). Sinon, votre travail ne sera pas corrigé. Le rapport doit obligatoirement inclure les éléments ou sections suivantes :

1. Page de présentation : elle doit contenir le libellé du cours, le numéro et l'identification du TP, le numéro de groupe, la date de remise, les matricules et noms des membres de l'équipe.
2. Introduction avec vos propres mots pour mettre en évidence le contexte et les objectifs du TP.
3. Présentation de vos travaux : une explication de votre solution. Ajoutez le diagramme de classes complet, contenant tous les attributs et toutes les méthodes ajoutées.
4. Difficultés rencontrées lors de l'élaboration du TP et les éventuelles solutions apportées.
5. Conclusion : expliquez en quoi ce laboratoire vous a été utile, ce que vous avez appris, vos attentes par rapport au prochain laboratoire, etc.

**Notez que vous ne devez pas mettre de code source dans le rapport.**

### 8.2 Soumission du livrable

La soumission doit se faire uniquement par Moodle.

## 9 évaluation

éléments évalués	Points
<b>Qualité du rapport</b> : respect des exigences du rapport, qualité de la présentation des solutions	2
<b>Qualité du programme</b> : compilation, structures de données, gestion adéquate des variables et de toute ressource (création, utilisation, libération), passage d'arguments, gestion des erreurs, documentation du code, etc.	2
<b>Composants implémentés</b> : respect des requis, logique de développement, etc.	
C1	2
C2	1.5
C3	1.5
C4	2
C5	1.5
C6	2
C7	2
C8	2
C9	1.5
Total de points	20

Notes

## 10 Documentation.txt

- <http://www.cplusplus.com/doc/tutorial/>
- [http://public.enst-bretagne.fr/\\$\sim\\$brunet/tutcpp/Tutoriel%20de%20C++.pdf](http://public.enst-bretagne.fr/$\sim$brunet/tutcpp/Tutoriel%20de%20C++.pdf)
- <http://fr.openclassrooms.com/informatique/cours/programmez-avec-le-langage-c>
- Algorithme de Dijkstra illustré : <https://www.youtube.com/watch?v=0nVYi3o161A>
- [learnxinyminutes.com](http://learnxinyminutes.com)

---

# Annexe

---

## 1 Format de fichier

- Le fichier Individus.txt comprend les individus et leurs tags respectifs selon le format suivant :  
individu1, tag1  
individu2, tag2
- Le fichier Contacts.txt comprend les individus et la distance qui a pu les séparer. Pour être exposé la distance de coût minimal doit au plus être égale à 2 mètres. Pour chaque *ligne<sub>i</sub>* du fichier on a : *individu<sub>i</sub> distance<sub>i</sub> voisin<sub>i</sub>*

## 2 Affichage du graphe

L'affichage du graphe se fait de la façon suivante :

*(individu<sub>i</sub> voisin<sub>i</sub> (distance entre eux))*

*(individu<sub>i</sub> voisin<sub>i</sub> (distance entre eux))*

...

## 3 Format messages alerte covid

- Aucune exposition détectée
- Vous avez été exposé au cours des 14 derniers jours