

Travail pratique #3 : Planification d'un réseau de panneaux publicitaires électroniques

1. Objectifs

- Utiliser des structures de données d'une bibliothèque normalisée (les conteneurs de la librairie standard de C++).
- Implémenter un type abstrait de données `Carte` basé sur la représentation de graphe.

2. Problématique

Vous devez écrire un programme en C++ nommé `tp3`. Le programme `tp3` doit chercher une solution au problème suivant.

La mairie d'une ville a décidé d'installer des terminaux publicitaires dans un quartier. Elle veut relier tous les sites (places publiques) du quartier entre eux via un réseau filaire. Elle vous a fourni les données suivantes:

- La liste des rues les plus visitées du quartier ;
- Le coût associé à l'installation des panneaux sur un segment de la rue.

Chaque rue est identifiée par un ensemble de sites. Les rues et les sites sont identifiés par les noms : `rue0 a b`. `rue0` est le nom de la rue, `a` et `b` sont les noms des sites.

Exemple : `rue0 a b 4`

Les sites du quartier sont reliés entre eux en formant un graphe connexe étiqueté où les nœuds représentent les sites du quartier, les arêtes sont les segments des rues qui relient les sites entre eux, et les poids sont les coûts sur chaque segment. La ville souhaite relier de manière optimale tous les sites du quartier par un réseau publicitaire en réduisant les dépenses d'installation. En cas d'arêtes de même poids, celles-ci doivent être traitées selon l'ordre alphanumérique des nœuds de départ. En cas d'égalité des poids et des nœuds de départ, utilisez l'ordre alphanumérique des nœuds d'arrivée.

3. Structure de programme

Pour bien amorcer ce travail, il est fortement recommandé de commencer avec le squelette de départ fourni dans `tp3.zip`. Certaines classes doivent être rajoutées à l'implémentation. Vous

pouvez modifier les fichiers du squelette autant que vous le désirez. Toutefois, pour la correction automatique, vous devez préserver la syntaxe d'appel du programme et ses formats d'entrée et de sortie.

3.1 Syntaxe d'appel

Le programme `tp3` doit pouvoir être lancé en ligne de commande avec la syntaxe suivante.

```
./tp3 carte.txt
```

Le fichier `carte.txt` spécifie une carte des sites d'un quartier et les données de coût associées à l'installation de panneaux sur chaque rue.

Les résultats produits par votre programme doivent être écrits dans la sortie standard (`stdout`) à l'aide du flux de sortie C++ `std::cout`.

3.2 Fichier `carte.txt`

Le fichier `carte.txt` est constitué de :

1. Une liste des sites (nœuds). Un site est spécifié par un nom (une chaîne de caractères);
2. Trois tirets (---) de séparation.
3. Une liste des rues d'installation de panneaux.
Une rue est spécifiée par un nom de rue (une chaîne de caractères); un deux-points (:); une paire de sites représentant un point de départ et un point d'arrivée d'une arête; un nombre caractérisant le coût d'installation sur ce segment de la rue (arête); un point-virgule (;).
4. Trois tirets (---) de fin.

À titre d'exemple, voici le fichier (`carte0.txt`) :

```
a
b
c
d
e
f
g
h
i
---
rue0 : a b 4 ;
rue1 : a h 8 ;
rue2 : b h 11 ;
rue3 : b c 8 ;
rue4 : c i 2 ;
rue5 : c f 4 ;
rue6 : c d 7 ;
```

```

rue7 : d f 14 ;
rue8 : d e 9 ;
rue9 : e f 10 ;
rue10 : f g 2 ;
rue11 : g i 6 ;
rue12 : g h 1 ;
rue13 : h i 7 ;
---
```

3.3 Format de sortie pour TP3

Le programme `tp3` doit afficher la sortie de la manière suivante.

D'abord, afficher les sites, un nœud par ligne. Ensuite, afficher les arêtes dans l'ordre croissant alphanumérique des nœuds de départ. Si plusieurs arêtes ont le même nœud de départ, vous devez afficher ces arêtes selon l'ordre croissant alphanumérique des nœuds d'arrivée.

Pour chaque arête vous devez afficher sur une ligne:

- Nom de la rue : `rue0`
- Sommets d'une arête : `a b`
- Étiquette : `4`

Après avoir affiché toutes les arrêtes :

1. Trois tirets (`---`) de séparation;
2. Le coût total d'installation des panneaux sur la dernière ligne.

Exemple de sortie.

```

a
b
c
d
e
f
g
h
i
rue0 a b 4
rue1 a h 8
rue6 c d 7
rue4 c i 2
rue8 d e 9
rue5 f c 4
rue10 g f 2
rue12 h g 1
---
37
```

4. Contraintes

4.1 Librairie standard C++ obligatoire

Pour ce `tp` vous devez utiliser, autant que possible, les conteneurs de la librairie standard de C++ (Standard Template Library). Cette contrainte vise à mettre en pratique l'utilisation d'une bibliothèque normalisée.

4.2 Environnement de développement

Relisez les Politiques et les directives sur les outils informatiques dans le cours INF3105.

4.3 Taille des équipes

Vous pouvez effectuer ce travail en équipe de 1 ou 2. Toutefois, tous les membres de l'équipe doivent contribuer à l'ensemble du travail et non à seulement quelques parties. Le travail d'équipe vise à favoriser les discussions et l'entraide. Le travail d'équipe ne vise pas à réduire la tâche. Ainsi, se diviser la tâche en deux n'est pas une méthode de travail d'équipe appropriée dans ce cours. Tous les membres de l'équipe doivent être en mesure de comprendre et d'expliquer l'ensemble du travail. La participation inadéquate d'une étudiante ou d'un étudiant peut être considérée comme du plagiat. Le professeur et le correcteur pourront sélectionner quelques équipes au hasard afin de vérifier que tous les membres sont capables d'expliquer l'ensemble du travail.

5. Tests

Quelques tests vous seront fournis. Vous pouvez également créer vos propres tests pour vérifier votre solution en profondeur. Le respect de la syntaxe d'appel et de la sortie est obligatoire. Votre note sera basée sur les résultats des tests que nous effectuerons.

6. Remise

Vous devez remettre le **tp3** au plus tard le 28 avril à 23h59.

6.1 Remise papier

La partie papier est constituée de:

1. Une page de présentation optionnelle. Si vous n'avez pas de page de présentation, indiquez clairement vos noms et codes permanents sur la première page.
2. Une analyse de la complexité algorithmique de votre solution.
 - Identifiez les principaux facteurs qui influencent le temps d'exécution.
 - Donnez la complexité temporelle en notation grand O, avec justification.

3. Code source. Imprimez le code source complet de votre programme. Remettre la partie papier dans la chute à travaux tout juste à côté du PK-4151.

6. 2 Remise électronique

Utilisez le système Moodle.

7. Évaluation

Ce travail pratique vaut 15% de la note finale.

Grille de correction

Critère	Description	Pondération
A.	Respect des directives pour la remise <ul style="list-style-type: none">Fichiers sources seulement (<code>makefile</code>, <code>.h</code>, <code>.cpp</code>). Aucun fichier source manquant. Aucun fichier intermédiaire (<code>.o</code>, <code>.obj</code>, <code>.gch</code>, etc.) ou exécutable (<code>tp3.exe</code>) n'a été remis. Aucun fichier de test n'a été remis.Remise par Moodle. Pas de remise par courriel.Compilable avec <code>make</code> sans modifications.Exécutable sans modification.	/ 2
B.	Appréciation générale Structure du programme. <ul style="list-style-type: none">Découpage du programme (tout n'est pas dans la fonction <code>main</code>).Justesse des types et des structures de données.Classes et fonctions.Usage du mot clé <code>const</code>.Usage des références et des pointeurs.Qualité du code.Nomination des identificateurs (noms significatifs), lisibilité du code, etc.Présence et pertinence des commentaires; etc.Encapsulation.Respect des principes de l'abstraction;Cachez le maximum de la représentation des objets en rendant un maximum d'attributs privés;	/ 2
C.	Fonctionnement correct. <ul style="list-style-type: none">Le programme produit les bons résultats.L'efficacité n'est pas directement évaluée. Cependant, l'efficacité peut être indirectement évaluée lorsqu'un programme ne parvient pas à produire des résultats dans des délais raisonnables.	/ 7

D.	Exactitude de l'auto-évaluation. <ul style="list-style-type: none"> Vous déclarez de quelle manière votre programme fonctionne : correctement, partiellement ou pas du tout. 	/ 1
E.	Analyse de l'algorithme Complexité temporelle en notation grand O. <ul style="list-style-type: none"> Ordre de grandeur simplifié. Ex: $O(2n) \implies O(n)$. Justification claire et correcte. 	/ 3
	Total :	15/ 15

Pour les cas problématiques, jusqu'à 2 points peuvent être retranchés pour la qualité de la présentation.