Rapport TP2 C - Prim

Corentin SABIER

11/02/2024

Table des matières

1 Description du code

1.1 Remarque sur le rendu

Tout d'abord, vous remarquerez que la qualité du travail pour ce TP est en-dessous de ce que j'ai rendu pour le premier TP. Je m'en excuse et je compte me mettre à niveau pour le dernier TP. Aussi je n'ai pas autant essayé de soigner les vérifications d'input utilisateurs, désolé si ça génère de la frustration pendant la navigation/correction.

1.2 Partis pris

Je vais succinctement décrire les libertés que j'ai pri.

1. Comme vu en présentiel, vous aviez parlé d'utiliser la pile pour implémenter notre solution. Étant pris par le temps j'ai décider de simplement résoudre le problème avec les solutions qui me venaient naturellement sans orienter ma réflexion vers l'utilisation d'une structure en particulier.

2. Je n'ai pas saisi/ pris le temps de comprendre comment implémenter une structure LH/LV pour les arbres générés par l'algorithme de Prim.

3. Ainsi, la structure de graphe que j'ai utilisé est la même pour un grpah quelconque et pour un arbre couvrant de poids minimal (**MST**).

1.3 Fonctionnement du menu

Comme pour le TP 1, le menu est composé de 6 options :

- **0. Charger un graphe ou afficher celui actuellement chargé**
- **1. Sauvegarder le graphe chargé**
- **2. Lancer Prim sur le graph chargé**

- **3. Charger un MST ou afficher l'arbre généré par Prim**
- **4. Sauvegarder le MST dans un fichier**
- **5. Quitter le programme**

1.4 Fonctionnalitées

Je ne vais expliciter ici que les fonctions qui pourraient contenir quelques ambiguïtés quant à leurs intéractions utilisateurs.

1.4.1 Charger/sauvegarder un graph/arbre

Ces fonctionnalités affiche automatiquement le graph s'il est chargé, la fonction prend en compte le chemin vers le repertoire *graph/* et ne se soucis pas de l'extension que vous pouvez mettre au fichier du graphe.

Si vous créez vos fichiers de graphes ou d'arbres. La seule différence utilisateurs entre un graphe et un arbre dans un fichier et la mention à la première ligne de "tree" **si c'est un arbre**.

Voici un exemple de fichier contenant un arbre :

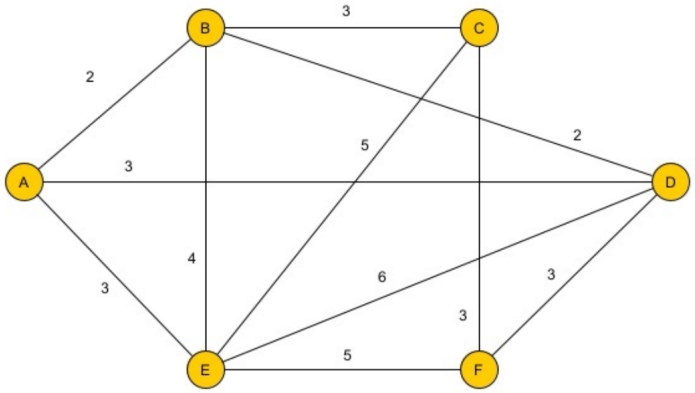
```
ABCDEF
AB2
AD3
AE3
BC3
BD2
BE4
CE5
CF3
DE6
DF3
EF5
```

Listing 1: graphTTP.txt

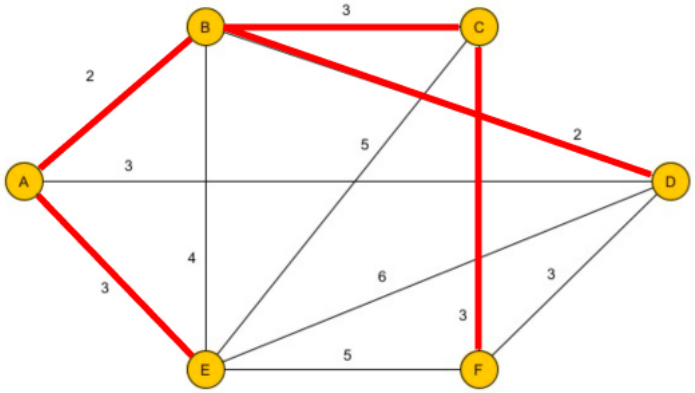
2 Jeux d'éssais

2.1 Test avec le graphe de l'énoncé

Ce test utilise le graph fourni dans l'énoncé du TP, soit :



On part du sommet **A** et est supposé obtenir l'arbre suivant :



et l'on l'arbre suivant avec notre programme :

```
Enter a number between 0 and 6: 3

Loading tree
The current graph is :

Vertices : ABDECF
Edges :
A <--> B   w = 2
B <--> D   w = 2
A <--> E   w = 3
B <--> C   w = 3
C <--> F   w = 3

```

2.1.1 Prim depuis F

on obtient bien l'arbre suivant :

```
Enter a number between 0 and 6: 3

Loading tree
The current graph is :

Vertices : FCBADE
Edges :
F <--> C  w = 3
C <--> B  w = 3
B <--> A  w = 2
B <--> D  w = 2
A <--> E  w = 3

```

2.2 Test avec un graph vide

Le graph ne contient que des sommets et aucune arête.

```
Enter a number between 0 and 6: 2

Getting minimal spanning tree (Using Prim)
The graph is empty or it doesn't have any edge, no MST can be found

```

2.3 Test avec un graph non connexe

Le graph contient deux composantes connexes. Il ne renverras l'arbre couvrant que de la composante connexe contenant le sommet demandé.

2.4 Test relance Prim

Cette fonctionnalité a été cassé par la modification à la dernière minute de la structure du graphe, plus exactement c'est le *freeGraph()* qui est cassé.

2.5 Test distance d'un point au sommet de l'arbre

Dans le premier MST (Sommet de départ A, graph de l'énoncé), la distance à F est :

```
Enter a number between 0 and 6: 5

Getting distance to a vertex (from the tree)
Enter the target vertex: F
The distance from the top to F is 8

```

2.6 Test distance d'un point qui n'est pas dans l'arbre

```
Enter a number between 0 and 6: 5

Getting distance to a vertex (from the tree)
Enter the target vertex: G
The vertex G is not in the tree

```