

Calcul d'arbre couvrant de poids minimum --- Algorithmes de Kruskal

Travail à effectuer en binôme.

Travail à effectuer

Vous devez mettre en œuvre un programme qui :

- lit les arêtes d'un graphe dans un fichier texte et en sauvegarde le contenu dans une structure de données de votre choix ;
- exécute l'algorithme de Kruskal pour calculer un arbre couvrant de poids minimum.

Vous pouvez choisir la méthode « constructive » ou la méthode « destructive ».

Votre code doit afficher les décisions prises par l'algorithme de Kruskal lors de son exécution :

- arête ajoutée ou supprimée à chaque étape ;
- arbre couvrant final (sous forme de matrice d'adjacence ou de liste des arêtes) et poids calculé.

Graphes

Il s'agit de graphes non orientés, valués et connexes.

Les sommets sont représentés par les nombres entiers de 0 à N-1 pour un graphe comportant N sommets.

Les valeurs associées aux arcs sont des nombres entiers quelconques.

Fichier de données

Basez-vous sur ce que vous avez fait pour les TP précédents.

Rappel – Méthodes de Kruskal

Soit $G=(S,A)$ pour lequel on recherche un arbre couvrant de poids minimal $G'=(S,A')$.

Kruskal constructif

Initialisation : $A' \leftarrow \emptyset$

Itérations : tant que G' n'est pas un arbre couvrant, on ajoute à G' des arêtes de poids le plus faible possible sans créer de cycle dans G'

Kruskal destructif

Initialisation : $A' \leftarrow A$

Itérations : tant que G' n'est pas un arbre, on supprime dans G' les arêtes de plus fort poids tout en gardant la connexité de G' .

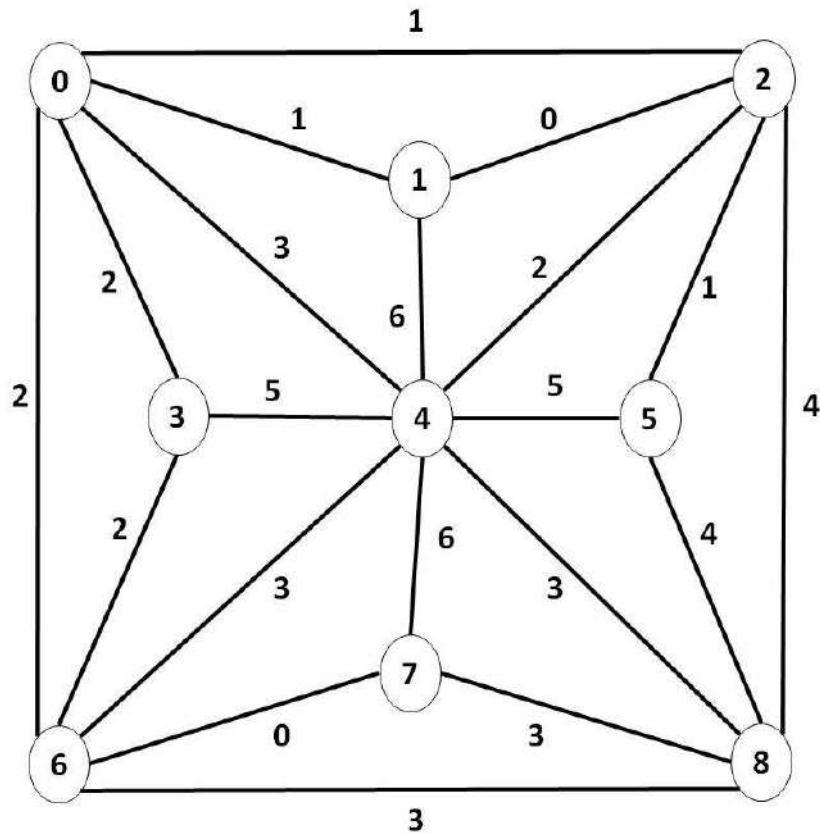
La méthode que vous déciderez de mettre en œuvre utilisera donc une des deux fonctions mises en œuvre au TP 2.

Langage de programmation

Comme pour les TP précédents.

Rendu du travail

Vous devrez exécuter votre programme sur le graphe ci-dessous :



Délai : 7 jours, samedi et dimanche inclus.

Envoi par email à la même adresse que les TP 1 et 2.

Sujet du mail : EFREI – L3 – TG – TP3 – *nom1 nom2*

Corps du message : vide

Pièces jointes :

Un fichier au format .txt contenant les traces d'exécution de votre programme.

Autant de fichier de code source (.c, .cpp, .h) que nécessaire.

Les fichiers de données utilisés par votre programme pour lire les graphes de test.

Pas d'archive. Pas de fichier binaire. Pas de « fichier projet » géré par votre outil de développement.

Tous les fichiers doivent avoir un nom préfixé par les noms des deux binômes, par exemple « barbot-lepoivre-tp3.cpp », barbot-lepoivre-tp3-graphe1.txt », ...

Tout manquement à ces consignes sera pénalisé.

Éléments de notation

Seront pris en compte :

- Votre code lui-même : choix des structures de données, algorithmes choisis et mise en œuvre de ces algorithmes.
- Son adaptabilité à d'autres graphes que ceux demandés en traces d'exécution.
- Les commentaires, et plus généralement la clarté de votre code.
- Les traces fournies.

Toute récupération évidente, même partielle, de code fait par un autre groupe se verra sanctionnée par un 0/20 à tous les étudiants concernés.