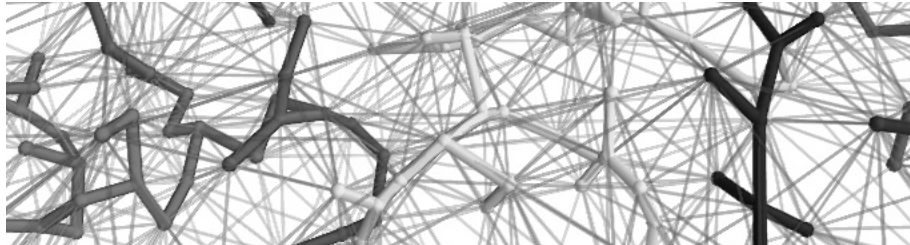


Projet – Minimum Spanning Tree

Ce projet est à réaliser par groupe de **2 étudiants maximum**.

Date limite de rendu : **vendredi 29 novembre minuit** au plus tard.

Règle : **4 points de moins par jour de retard**.



Objectif :

L'objectif de ce projet est d'implémenter les algorithmes de construction d'arbres recouvrants de poids minimum (**Minimum Spanning Tree - MST**) vus en cours ; et également de mettre en œuvre au moins un algorithme pour résoudre le problème d-MST' (**Degree-Constrained Minimum Spanning Tree**) qui impose une contrainte de degré maximum sur l'arbre construit.

Dans un premier temps, vous devez, en utilisant les classes que je vous ai données :

1. Implémenter les deux versions de l'**algorithme de Kruskal** et l'**algorithme de Prim** (prendre soin d'utiliser des structures de données adaptées). Attention à la complexité de vos algorithmes.
2. Tester tout d'abord ces algorithmes sur de « petits » graphes (ceux donnés en cours par exemple).
3. Tester ensuite ces algorithmes sur les graphes que vous trouverez dans le répertoire « Projet » sur Claroline.
4. Réaliser une étude comparative de ces algorithmes en termes de temps de calculs et de résultats.

Dans le problème d-MST, le degré maximum de l'arbre recouvrant ne doit pas dépasser d . Dans ce cas :

5. Rechercher ou imaginer un algorithme de construction d'un arbre avec cette contrainte de degré maximum à ne pas dépasser. Il s'agit ici de construire un algorithme dit « constructif » utilisant une heuristique de votre choix (le résultat optimal ne sera donc pas garanti).
6. Réaliser des tests avec différentes valeurs de d sur les graphes de Claroline. En particulier vous afficherez le **rapport « Poids de l'arbre d-MST » sur « Poids de l'arbre MST » en fonction de la valeur de d** .

Vous devez fournir un rapport en PDF et fournir le code associé. Tout ceci devra être déposé dans un ZIP à votre nom dans la « zone de dépôt » du module Claroline associé à votre cours.

L'ensemble des algorithmes et méthodes développés devront être expliqués clairement (en particulier l'algorithme de la question 5). Vos algorithmes devront être testés et illustrés (tableaux, graphiques, ...) sur des exemples.