Fiche d'exercices pratiques 4 - 3h

Consignes

Les consignes de la première fiche de TP restent valides.
Attention : vous devez déposer votre travail à la fin des deux séances consacrées à cette fiche sur `Moodle` https://moodle.uphf.fr/course/view.php?id=1681

1. Le travail en binôme est autorisé.
2. N'oubliez pas d'indiquer vos nom(s) et prénom(s) en commentaire en haut de votre (vos) fichier(s).
3. Ne déposez que les fichiers sources (code) de votre travail.
4. Ne négligez pas les commentaires pour expliquer vos choix et les fonctions mises en œuvre.

Le respect des consignes sera pris en compte dans la notation.
La version déposée à la fin de la séance sera celle évaluée. Néanmoins une version améliorée / complétée pourra être déposée au plus tard une semaine après la séance, et pourra être prise en compte sous la forme d'un *bonus* dans la notation.

Exercice 1 : arbre couvrant

On considère une usine d'assainissement qui doit déployer un réseau de collecte et de distribution d'eau à un groupe de maisons, via des "tuyaux". Le réseau est organisé de sorte qu'une maison peut être soit connectée par un ensemble de tuyaux allant directement à l'usine, soit indirectement par un tuyau reliant la maison à une autre maison voisine, elle-même ayant une connexion menant à l'usine. On pourrait par exemple avoir le réseau suivant, composé des maisons `A, B, C` et d'une usine :

$$\texttt{A} \longleftrightarrow \texttt{B} \longleftrightarrow \texttt{C} \longleftrightarrow \texttt{usine}.$$

Un coût est associé à chaque tuyau et la société gérant l'usine souhaite minimiser le coût total du réseau à mettre en œuvre.

On peut modéliser ce problème via un graphe non orienté dans lequel les sommets sont les maisons et l'usine, alors que les connexions possibles sont représentés via des arêtes pondérées (avec le coût de mise en œuvre du tuyau correspondant). Le problème revient alors à chercher un arbre couvrant de poids minimum dans le graphe. Il s'agit donc de définir un sous-ensemble d'arêtes permettant d'avoir une couverture de l'ensemble des sommets (i.e., un graphe connexe), et tel que la somme des coûts des arêtes sélectionnés est la plus petite possible.

On peut résoudre ce problème via plusieurs algorithmes. Dans ce TP on va s'intéresser à l'algorithme de Prim décrit dans l'Algorithme 1.

Algorithme 1 : Algorithme de Prim

$T \leftarrow \emptyset$;
$\overline{S} \leftarrow \{s\}$;
tant que $\left|\overline{S}\right| \neq n$ **faire**
  Trouver $a = (y, x) \in A$ de poids minimal tel que $x \in \overline{S}$ et $y \in S - \overline{S}$;
  $T \leftarrow T \cup \{a\}$;
  $\overline{S} \leftarrow \overline{S} \cup \{y\}$;
fin

Cet algorithme prend en entrée le graphe $G = (S, A)$ et un sommet source, s. Il cherche à construire l'arbre T en y ajoutant à chaque itération une nouvelle "branche" parmi les arêtes joignant un sommet dejà dans T et un sommet pas encore couvert. L'ensemble S permet de savoir quels sommets n'ont pas encore été ajoutés.

1. Choisir une des représentations d'un graphe des TP précédents et l'adapter légèrement pour prendre en compte la présence des coûts sur les arêtes.
2. Implémenter l'algorithme, en prenant en compte qu'il faut retourner l'arbre ainsi construit. Votre solution affichera l'arbre obtenu ainsi que le coût associé.
3. Vérifier le bon fonctionnement de l'algorithme sur quelques graphes générés aléatoirement (vous pouvez reprendre et adapter l'algorithme de l'exercice 2 du TP 1 pour associer un coût généré aléatoirement dans un intervalle, par exemple $[1, 10]$, pour chaque arête).