Algorithmique II
Graphes – Arbres

Exercice 1 :

Quizz : vrai ou faux avec justifications ! Dans tout ce qui suit MST ("Minimum Spanning Tree") désignera un arbre couvrant de poids minimum d'un graphe connexe donné.
Donner une réponse justifiée (preuve directe ou preuve algorithmique) aux questions suivantes.

1. Est-il vrai que dans un graphe pondéré g, l'arête de poids minimum si elle est unique est toujours incluse dans le MST de g ?
2. Qu'en est-il si cette arête n'est pas unique ?
3. Dans un graphe ou toutes les arêtes ont des poids 2 à 2 distincts on a un seul et unique MST.
4. Soit e une arête de poids maximum sur un des cycles de $G = (S, A)$. Soit $G' = (S, A - \{e\})$. Est-il vrai que $\mathrm{MST}(G') = \mathrm{MST}(G)$?
5. Les algorithmes de Borůvka, Kruskal et Prim continuent à trouver les MST sur des graphes avec des arêtes de poids négatifs.

Exercice 2 :

Ecrire un algorithme qui trouve l'arbre couvrant de poids *maximum* d'un graphe pondéré. Quelle est sa complexité ?

Exercice 3 :

Etant donné un graphe pondéré g et un ensemble d'arêtes $\mathcal{E} = \{e_1, e_2, \cdots\}$ de g, écrire un algorithme qui calcule le MST de g incluant forcément $\mathcal{E}$.

Exercice 4 :

Etant donné un graphe pondéré g, on sait calculer son MST via différents algorithmes. Ecrire un algorithme dit de mise-à-jour qui calcule un (nouveau ?) MST si le poids d'une arête (u, v) de g est modifié. Quelle est la comlexité de votre algorithme ? Votre graphe g s'est enrichi avec un nouveau sommet v et de nouvelles arêtes $a_1, a_2, \cdots$. Ecrire un algorithme de mise-à-jour du nouveau MST.

Exercice 5 :

Une arête du MST est dit *critique* si sa suppréssion et son remplacement dans le MST augmente le poids de ce dernier. Ecrire un algorithme qui trouve toutes les arêtes critiques. Quelle est sa complexité ?

Exercice 6 :

Le problème dit du *voyageur de commerce* ("Traveling Salesman Problem" ou TSP) consiste à trouver un tour (un cycle) de poids minimum entre n villes. Etant donné n villes et les coûts de voyage entre chaque pair de ville $c(v_i, v_j)$.

1. Montrer sur un exemple qu'il y a un nombre exponentiel de solutions potentielles au problème du TSP.
2. Donner un algorithme qui donne une borne inférieure du coût du tour minimum.