AL5
TD n° 8 : Arbres Couvrants Minimaux, algorithme de Prim

Exercice 1 : algorithme de Prim

Appliquer l'algorithme de Prim sur le graphe ci-dessous. On représentera la file de priorité à chaque itération.

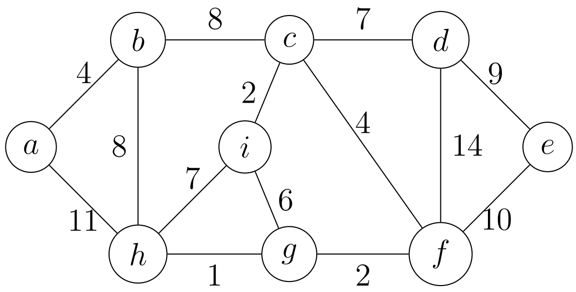

FIGURE 1 – graphe G pour appliquer l'algorithme de Prim

Exercice 2 : vrai ou faux

Soit $G = (V, E)$ un graphe connexe avec une pondération $w : E \to \mathbb{R}$. Décidez pour chacune des affirmations suivantes si elle est vraie ou non. Justifiez vos réponses (si l'assertion est vraie, prouvez-la, sinon donnez un contre-exemple).

1. Si le graphe G a plus de $|V| - 1$ arêtes, et qu'il existe une unique arête de poids maximum, alors cette arête ne peut pas faire partie d'un ACM de G.
2. Soit e une arête de poids minimum dans G. Alors, il existe un ACM de G comprenant l'arête e.
3. Si l'arête de poids minimum de G est unique, alors elle doit faire partie de chaque ACM de G.
4. Si e appartient à un ACM de G, alors e est une arête de poids minimum d'une coupe de G.
5. Soit C un cycle dans un graphe G ayant une unique arête e de poids minimum. Alors e appartient à chaque ACM de G.
6. Soit C un cycle dans un graphe G ayant une unique arête e de poids maximum. Alors e appartient à chaque ACM de G.
7. Le plus court chemin entre deux sommets de G appartient nécessairement à un ACM de G.
8. L'algorithme de Prim fonctionne correctement lorsqu'il y a des arêtes de poids négatif.

Exercice 3 : ACM vs PCC : le cas “max”

Ici on souhaite modifier les algorithmes de Dijkstra et de Prim pour essayer de calculer un arbre couvrant maximal (pour Prim) et les plus longs chemins depuis un sommet vers tous les autres (pour Dijkstra). Les modifications qu'on fait sont très simples :

- on utilise un tas prioritaire “max” (au lieu de “min”) ;
- on initialise le tableau de distance avec $-\infty$ (au lieu de ∞) ;
- quand on examine $d(b) \overset{(?)}{\lesseqgtr} d(a) + w(a,b)$ pour Dijkstra, on met à jour $d(b)$ si la nouvelle distance passant par a est plus grande que la distance sauvegardée ;
- de même pour Prim, quand on examine $d(b) \overset{(?)}{\lesseqgtr} w(a,b)$, on met à jour $d(b)$ si $w(a,b) > d(b)$.

1. L'algorithme de Prim ainsi modifié calcule-t-il un arbre couvrant maximal ?

2. L'algorithme de Dijkstra ainsi modifié calcule-t-il les longueurs des plus longs chemins depuis un sommet s ?

Exercice 4 : Caractérisations des ACM et "Kruskal à l'envers"

Soit $G = (V, E, w)$ un graphe connexe pondéré. Soit T un arbre couvrant de G. On note $E(T)$ l'ensemble des arêtes de T.

1. Montrer que si $e \in E \setminus E(T)$, il existe un unique cycle (que l'on notera par la suite C_e) dans le graphe $(V, E(T) \cup \{e\})$.

2. Montrer que les propositions suivantes sont équivalentes :

(i) T est un ACM de G.

(ii) $\forall e \in (E \setminus E(T))\ \forall e' \in C_e\ \ w(e') \leqslant w(e)$.

(iii) Pour **toute** partition $(U, V \setminus U)$ de l'ensemble des sommets, il existe une arête de T qui traverse la partition et qui est de poids minimum parmi toutes les arêtes reliant un sommet de U à un sommet de $V \setminus U$

Indication : montrer $(i) \implies (ii) \implies (iii) \implies (i)$. Pour $(iii) \implies (i)$, on pourra montrer qu'il existe un déroulement de l'algorithme de Prim qui produit exactement T (meme si on peut faire sans cet argument)

3. Si e est une arête de G, on note $G - e$ le graphe pondéré obtenu en supprimant l'arête e et en gardant les même poids pour les autres arêtes. En utilisant la question 2, montrer que si e est une arête de poids maximum dans G telle que le graphe $G - e$ est encore connexe, alors tout ACM de $G - e$ est aussi un ACM de G.

4. Déduire de la question précédente un algorithme qui part d'un graphe G et construit un ACM en retirant une à une des arêtes du graphe. Sa complexité est-elle aussi bonne que celle de Prim ?