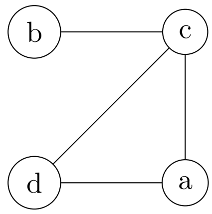
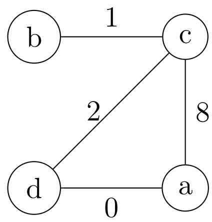
AL5
TD n^o 7 : arbres couvrants minimaux

Exercice 1 : Arbre couvrant sur un graphe à quatre arrêtes

1. Expliquer ce qu'est :
   – Un arbre couvrant d'un graphe non orienté et connexe. Donner les deux (seuls) arbres couvrants pour le graphe suivant :



   – Arbre couvrant de poids minimal d'un graphe non orienté connexe, dont les arrête sont pondérées. Déterminez celui du graphe pondéré non orienté, connexe, suivant :

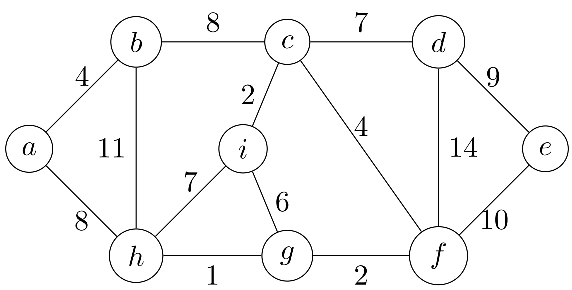


2. Pourquoi le graphe est-il connexe ?

Dans le cas général, on ne peut pas déduire aussi facilement et à la main l'arbre couvrant, ou énumérer tous les arbres couvrants. Voyons donc dans le prochain exercice comment calculer un arbre couvrant de poids minimal.

Exercice 2 : algorithmes de Kruskal et de Prim

Trouver un arbre couvrant de poids minimal sur le graphe suivant en appliquant les algorithmes de Kruskal et de Prim. Pour l'algorithme de Prim, on représentera la file de priorité à chaque itération.



Exercice 3 : Unicité de l'arbre couvrant de poids minimal

Montrer que si tous les poids des arrêtes d'un graphe connexe pondéré sont différents, alors l'arbre couvrant de poids minimal est unique. Donnez un contre-exemple dans le cas contraire.

Exercice 4 : Ordre de parcours

Selon l'ordre dans lequel les arêtes de même poids sont examinées, l'algorithme de Kruskal ne retourne pas le même arbre couvrant minimal. Pour un arbre couvrant donné, existe-t-il nécessairement un ordre tel que cet arbre soit la sortie de l'algorithme de Kruskal ?

Exercice 5 : 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. Si e appartient à un ACM de G, alors e est une arête de poids minimum d'une coupe de G.
3. Soit C un cycle dans un graphe G ayant une unique arête e de poids minimum. Alors e appartient à chaque ACM de G.
4. Soit C un cycle dans un graphe G ayant une unique arête e de poids maximum. Alors e appartient à chaque ACM de G.
5. Le plus court chemin entre deux sommets de G appartient nécessairement à un ACM de G.
6. L'algorithme de Prim fonctionne correctement lorsqu'il y a des arêtes de poids négatif.

Exercice 6 : 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 $d(b) \overset{(?)}{\leqslant} d(a) + w(a, b)$ 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 $d(b) \overset{(?)}{\leqslant} w(a, b)$, 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 ?