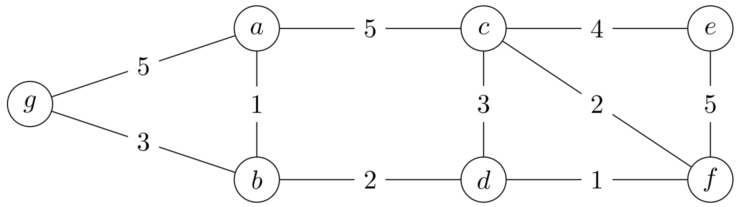
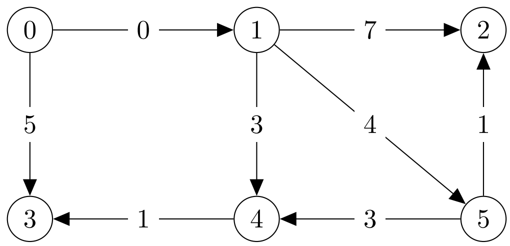
I Application des algorithmes

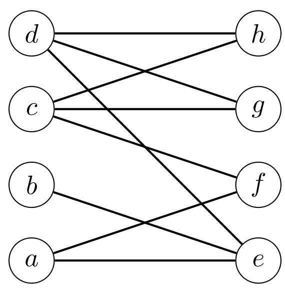
1. Appliquer l'algorithme de Kruskal à la main sur le graphe suivant :



2. Appliquer l'algorithme de Dijkstra à la main pour trouver les distances depuis let sommet 0 vers n'importe quel autre sommet :



3. Appliquer l'algorithme du cours pour trouver un couplage maximum dans le graphe suivant :



II Plus courts chemin et Dijkstra

1. Trouver un exemple de graphe pour lequel Dijkstra ne fonctionne pas (ne donne pas les plus courts chemins).
2. Change t-on les plus courts chemins si on additionne/multiplie les poids de toutes les arêtes d'un graphe par une constante?
3. Le graphe orienté $\overrightarrow{G} = (V, \overrightarrow{E})$ d'une chaîne de Markov possède une probabilité sur chaque arc. La probabilité d'un chemin est le produit des probabilités de ses arcs. Comment trouver un chemin le plus probable d'un sommet à un autre?
4. Soit $\overrightarrow{G}$ un graphe sans arc de poids négatif et s, t deux sommets. Montrer que l'on peut trouver tous les arcs utilisés par au moins un plus court chemin de s à t, en utilisant 2 fois l'algorithme de Dijkstra.
5. On suppose avoir un graphe orienté $\overrightarrow{G}$ dont les **sommets** sont pondérés par w, au lieu des arêtes. Le poids d'un chemin est alors la somme des poids de ses sommets. Comment modifier $\overrightarrow{G}$ pour pouvoir trouver des chemins de plus petit poids de $\overrightarrow{G}$, avec les algorithmes du cours?

III Plus large chemin

Soit $G = (V, E)$ un graphe pondéré. La **largeur** $l(C)$ d'un chemin est le minimum des poids de ses arêtes. Soient $u, v \in V$. Un chemin de u à v est un **plus large chemin** (*widest path*) s'il n'existe pas d'autre chemin de u à v de plus grande largeur.

1. Donner un plus large chemin de g à f dans le graphe du I.1.
2. Comment modifier l'algorithme de Kruskal de façon à trouver un arbre couvrant T de poids maximum dans G ?
3. Soit C le chemin de u à v dans T. Montrer que C est un plus large chemin de u à v (l'algorithme de Kruskal permet donc de trouver les plus larges chemins).

IV Questions sur les arbres couvrants de poids minimum

Soit $G = (V, E)$ un graphe pondéré.

1. Soit C un cycle de G et $e = \{u, v\}$ une arête de C dont le poids est strictement supérieur au poids des autres arêtes de C. Montrer que e ne peut pas appartenir à un arbre couvrant de de poids minimum de G.
2. (Propriété d'échange) Soient T_1, T_2 deux arbres couvrants de G et e_2 une arête de $T_2 - T_1$. Montrer qu'il existe une arête e_1 de T_1 telle que $T_1 - e_1 + e_2$ (le graphe obtenu en remplaçant e_1 par e_2 dans T_1) est un arbre couvrant de G.
3. Montrer que si tous les poids des arêtes de G sont différents, alors G admet un unique arbre couvrant de poids minimum.
4. Le nombre de domination $d(G)$ d'un graphe $G = (V, E)$ est le cardinal minimum d'un ensemble $S \subseteq V$ tel que $\forall v \in V$, $v \in S$ ou v est adjacent à un sommet de V'.
   Montrer que si G est connexe alors $d(G) \geq \dfrac{|V|}{2}$.
5. Soit T_1 un arbre couvrant de poids minimum de G et T_2 le 2ème plus petit arbre couvrant, c'est-à-dire l'arbre couvrant de poids minimum en excluant T_1. Montrer que T_1 et T_2 diffèrent d'une arête et en déduire un algorithme pour trouver T_2.

V Mise à jour d'arbre couvrant de poids minimum

Soit $G = (V, E)$ un graphe pondéré et T un arbre couvrant de poids minimum de G. Soit $e \in E$.

1. On diminue le poids de e. Expliquer comment mettre à jour T pour qu'il soit toujours un arbre couvrant de poids minimum.
2. On augmente le poids de e. Expliquer comment mettre à jour T pour qu'il soit toujours un arbre couvrant de poids minimum.

VI Hypercube

Un **hypercube** Q_n a pour sommets les mots binaires de taille n, 2 sommets étant reliés s'ils différent d'un bit.

1. Dessiner Q_3.
2. Quel est le nombre de sommets et d'arêtes de Q_n?
3. Montrer que Q_n est biparti.
4. Montrer que Q_n possède un couplage parfait.
5. Soit $n \geq 2$. Montrer que Q_n est **hamiltonien** : il existe un cycle qui visite tous les sommets exactement une fois. Dessiner un tel cycle de Q_3.

VII Questions sur les couplages

1. Soit G un graphe. Montrer que si G a un couplage parfait alors G possède un nombre pair de sommets. La réciproque est-elle vraie ?
2. Soit M_1 et M_2 deux couplages d'un graphe G, avec M_2 maximal. Montrer que $|M_1| \leq 2|M_2|$, puis donner un cas d'égalité.
3. Soit $G = (V, E)$ un graphe. Une couverture par sommets (*vertex cover*) de G est un ensemble C de sommets tels que chaque arête de G est adjacente à au moins un sommet de C. L'objectif est de trouver une couverture par sommets C^* de cardinal minimum.
   On propose l'algorithme suivant :

   2-approximation de vertex cover

   $M \longleftarrow$ couplage maximal de G
   $C \longleftarrow$ ensemble des sommets couverts par M

   Montrer que C est bien un couplage et que $|C| \leq 2|C^*|$.

4. Soit $M = (m_{i,j})$ une matrice de taille $n \times n$. On définit le permanent de M :

$$\mathrm{per}(M) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} m_{i,\sigma(i)}$$

où S_n est l'ensemble des permutations de $\{1, \ldots, n\}$.
Définir un graphe G dont le nombre de couplages parfaits est égal à $\mathrm{per}(M)$.
Remarque : il n'existe pas d'algorithme efficace pour calculer le permanent d'une matrice, contrairement au déterminant qui peut être calculé en $O(n^3)$ avec l'algorithme du pivot de Gauss.

VIII Questions sur les graphes bipartis

1. Montrer qu'un arbre est un graphe biparti.
2. Montrer qu'un graphe est biparti si et seulement s'il n'a pas de cycle impair.