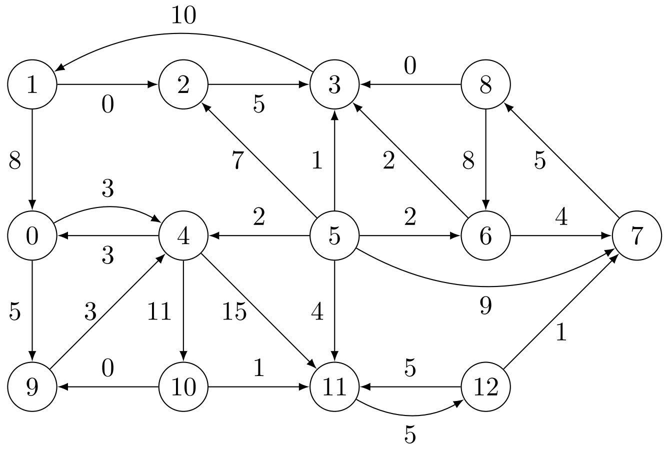
Fiche TD16 : Algorithmique des graphes

Exercice 1 *Quelques gammes*

On considère le graphe G suivant :



1. Appliquer l'algorithme de Kosaraju à ce graphe (en indiquant les dates de début et de fin de traitement de chaque sommet lors du premier parcours ; dès qu'un choix se présente dans ledit parcours, on considèrera en premier le sommet de plus petit numéro possible).
2. Calculer le graphe quotient associé à G.

On considère à présent le graphe G', égal à G dans lequel on a supprimé toutes les orientations (ce qui est possible car à chaque fois qu'il existe un couple d'arêtes orientées (u, v) et (v, u), ces deux arêtes ont le même poids).

3. Calculer les plus courts chemins à origine le sommet 1 (ainsi que leurs poids) grâce à l'algorithme de Dijkstra.
4. Déterminer un arbre couvrant minimal pour G' à l'aide de l'algorithme de Kruskal ainsi que son poids.

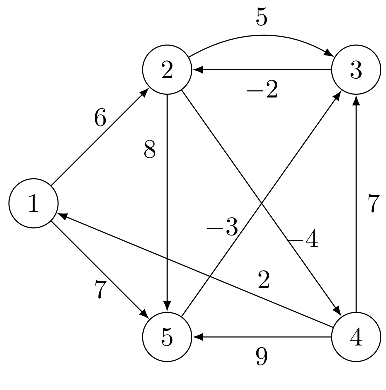
Exercice 2 *Algorithme de Bellman-Ford*

On considère l'algorithme suivant du à Bellman et Ford :

Entrée : Un graphe $G = (S, A)$ pondéré par une fonction w et un sommet $s \in S$.
Sortie : Un tableau D contenant à la case i le poids d'un plus court chemin de s à i si G n'admet pas de cycle de poids strictement négatif accessible depuis s et la phrase "il y a un cycle de poids < 0" sinon.

Bellman_Ford(G, s) =
  $D \leftarrow$ tableau de $|S|$ cases contenant $+\infty$ partout sauf $D[s]$, qui contient 0
  Pour k = 1 à $|S| - 1$
    Pour toute arête $(u, v) \in A$
      $D[v] \leftarrow \min(D[v], D[u] + w(u, v))$
  Pour toute arête $(u, v) \in A$
    Si $D[u] + w(u, v) < D[v]$
      Renvoyer "il y a un cycle de poids < 0"
  Renvoyer D

1. Appliquer cet algorithme au graphe suivant avec $s = 1$:



Que se passerait-il si l'arc $(4, 5)$ était de poids 8 au lieu de 9 ?

Dans toute la suite, afin de clarifier le propos, on utilisera la notation suivante : pour tout sommet $v \in S$ et tout $k \in [\![0, |S|-1]\!]$, $d(v, k)$ désigne le contenu de la case $D[v]$ à la fin de la k-ème itération de la première boucle pour ($d(v, 0)$ est donc le contenu de la case $D[v]$ avant de rentrer pour la première fois dans la boucle).

2. Montrer que s'il existe un cycle de poids strictement négatif dans G alors l'algorithme le détecte. *Indication : On pourra raisonner par l'absurde.*

3. Montrer que lorsque G n'admet aucun cycle de poids strictement négatif, pour tout $k \in [\![0, |S|-1]\!]$ et tout $v \in S$, $d(v, k)$ est le poids d'un plus court chemin de s à v empruntant au plus k arcs.

4. Déduire de ce qui précède que l'algorithme de Bellman-Ford est correct vis à vis de sa spécification.

5. Quelle est la complexité de cet algorithme ?

Remarque : L'algorithme de Bellman-Ford consiste en fait à relâcher $|S|-1$ fois toutes les arêtes du graphe. Après ces relâchements, on obtient un certain tableau, et si un relâchement de plus modifie ce tableau (ce que teste la deuxième boucle pour), c'est le signe d'un cycle de poids strictement négatif.

Exercice 3 *Plus courts chemins et multiplications de matrice*

On considère un graphe pondéré par w, à n sommets et sans cycle de poids strictement négatif et M sa matrice d'adjacence (la case (i, j) contient le poids de l'arête (i, j) si elle existe, 0 si $i = j$ et $+\infty$ si aucun chemin ne mène de i à j.). On définit la suite de matrices suivantes :

$D_1 = M$ et pour tout $m \geq 2$, $D_m = (d_{i,j}^m)_{1 \leq i,j \leq n}$ avec $d_{i,j}^m = \min\{d_{i,k}^{m-1} + w(k, j) \mid k \in [\![1, n]\!]\}$

1. Montrer que $d_{i,j}^m$ est le poids d'un plus court chemin de longueur au plus m entre les sommets i et j.

2. Décrire un algorithme de calcul de plus courts chemins pour tous couples de sommets utilisant ce résultat.

3. Quelle est la complexité naïve de cet algorithme ? La raffiner autant que possible.

Exercice 4 *Plus courts chemins dans un graphe acyclique orienté*

On propose l'algorithme suivant :

Entrée : Un graphe $G = (S, A)$ acyclique, orienté, pondéré par w et un sommet $s \in S$.
Sortie : Un tableau D contenant à la case i la quantité $\delta(s, i)$.
PCC_DAG(G,s) =
  Trier G topologiquement
  $D \leftarrow$ tableau de $|S|$ cases contenant $+\infty$ sauf $D[s]$ qui contient 0
  Pour chaque sommet u pris dans l'ordre topologique
    Pour chaque voisin v de u
      $D[v] \leftarrow \min(D[v], D[u] + w(u, v))$
  Renvoyer D

1. Si u est accessible depuis s, il existe un plus court chemin $s = s_0, ..., s_k = u$ entre s et u. Puisqu'on a trié topologiquement les sommets, l'arête (s_0, s_1) sera la première arête de ce chemin à être relâchée puis (s_1, s_2)... et enfin (s_{k-1}, s_k). Montrer que pour tout $i \in [\![0, k]\!]$, $D[s_i] = \delta(s, s_i)$.

2. En déduire la correction de cet algorithme.

3. Quelle est sa complexité ? La comparer aux complexités des algorithmes de calcul de plus courts chemins à origine unique que vous connaissez.

Remarque : Une application de cet algorithme est le calcul de chemins critiques (ou de leur poids) dans un diagramme PERT, ce qui permet de déterminer le temps minimal requis pour effectuer un ensemble de tâches dont certaines dépendent de la réalisation d'autres.

Exercice 5 *Deuxième arbre couvrant le moins lourd*

On considère un graphe $G = (V, E)$ *(utilisons les notations anglaises pour changer !)* non orienté, connexe et pondéré par une fonction $w : E \to \mathbb{R}$. On considère deux arbres couvrants minimaux de ce graphe, (V, A) et (V, A') qu'on suppose distincts. Dans ces conditions, l'existence d'une arête $e \in A\Delta A'$ de poids minimal est garantie.

1. Supposons par exemple que $e \in A \setminus A'$. Montrer alors qu'il existe une arête $e' \in A' \setminus A$ telle que $w(e) = w(e')$. *Indication : étudier le graphe $G' = (V, E \setminus \{e\})$.*

2. En déduire que si w est injective, alors G admet un unique arbre couvrant minimal.

On suppose par la suite que w est injective. Le graphe G admet donc un unique arbre couvrant minimal. On note (V, A) l'arbre couvrant minimal de G et (V, B) un second arbre couvrant par ordre de poids croissant.

3. Montrer que le second arbre couvrant par ordre croissant de poids n'est pas unique en exhibant un exemple.

4. Montrer qu'il existe $e \in A$ et $e' \notin A$ tel que $B = (A \setminus \{e\}) \cup \{e'\}$.

5. Si (V, C) est un arbre couvrant quelconque, pour tout $(u, v) \in V^2$, on note $\max_C(u, v)$ une arête de poids maximal sur le chemin unique reliant u à v dans (V, C). Décrire un algorithme qui permet de calculer les valeurs de $\max_C(u, v)$ pour tous les couples $(u, v) \in V^2$ en temps quadratique en $|V|$.

6. En déduire un algorithme - dont on déterminera la complexité - permettant de calculer un arbre couvrant de G ayant le deuxième poids le plus gros parmi tous les arbres couvrants G.

Exercice 6 *Approximation pour le problème du voyageur de commerce*

On considère un ensemble $V = \{v_1, ..., v_n\}$ de villes et d une distance. On suppose que pour tout $i \in [\![1, n]\!]$, il existe une route (non orientée) entre v_i et v_j : le graphe modélisant la situation est donc un graphe complet G tel que l'arête (v_i, v_j) est pondérée par $d(v_i, v_j)$. On cherche à construire un chemin dans G qui passe par chacune des villes et revient à son point de départ en minimisant la distance parcourue. Il s'agit donc de trouver un cycle couvrant de poids minimal dans G.

1. Montrer que si C_m est un cycle couvrant de poids minimal dans G, on peut supposer que C_m passe une et une seule fois par chaque sommet de G.

2. Soit A un arbre inclus dans G et C le cycle décrit par un parcours en profondeur de A. Montrer que $d(C) \leq 2d(A)$ où $d(X)$ désigne la somme des distances (poids) de toutes les arêtes de X.

3. En déduire que, si C est le cycle décrit par le parcours en profondeur d'un arbre couvrant minimal de G et C_m est un cycle couvrant minimal dans G, alors $d(C) \leq 2d(C_m)$.

4. Quelle est la complexité du calcul d'un cycle C tel que dans la question 4 en fonction du nombre de villes ?

Remarque : Le problème décrit dans cet énoncé est celui du voyageur de commerce. C'est un problème célèbre, notoirement NP-complet. L'exercice montre néanmoins qu'on peut trouver en temps polynomial une solution approchée pour ce problème qui est au pire 2 fois plus grosse qu'une solution optimale.

Exercice 7 *Couplages parfaits dans les graphes bipartis*

Une compagnie aérienne dispose de 5 avions a_1, a_2, a_3, a_4, a_5 qu'elle fait voler sur 5 lignes aériennes l_1, l_2, l_3, l_4, l_5. Tous les avions ne peuvent pas voler sur toutes les lignes, plus précisément : a_1 ne peut assurer que la liaison l_1, a_2 les liaisons l_1 et l_2, a_3 les liaisons l_1, l_3, l_4, l_5, a_4 les liaisons l_3, l_4, l_5 et a_5 les liaisons l_1, l_2.

1. Dessiner le graphe biparti modélisant la situation.

2. Déterminer un couplage maximum dans ce graphe grâce à l'alorithme de Berge et en déduire combien de liaisons la compagnie peut assurer simultanément au maximum. Le couplage trouvé est-il parfait ?

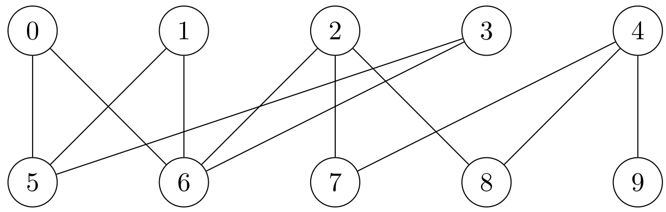
Soit G un graphe biparti dont on note (U, V) une bipartition des sommets. On cherche à montrer le théorème de Hall : Il existe un couplage de taille $|U|$ dans G si et seulement si, pour tout sous-ensemble $X \subset U$, l'ensemble $V(X)$ des voisins de X vérifie $|V(X)| \geq |X|$.

3. Montrer le sens direct de ce théorème.

4. En s'aidant au besoin d'un récurrence sur la taille de $|U|$, montrer le sens réciproque.

5. En déduire que G admet un couplage parfait si et seulement si $|U| = |V|$ et pour tout sous ensemble $X \subset U$, $|V(X)| \geq |X|$.

6. Le graphe biparti suivant admet-il un couplage parfait ? En trouver un couplage maximum en justifiant.



7. Montrer que si G est un graphe biparti k-régulier avec $k \geq 1$ (c'est-à-dire, tous les sommets de G ont degré égal à k), alors G admet un couplage parfait. *Indication : Utiliser la question 5.*

Remarque : Le théorème de Hall peut aussi s'énoncer avec le vocabulaire de la théorie des ensembles. On le trouve parfois sous le nom de lemme des mariages : alors, U représente un ensemble de femmes, V un ensemble d'hommes, une arête entre u et v indique que u et v sont d'accord pour se marier et le théorème de Hall donne une condition nécessaire et suffisante pour que toutes les femmes puissent se marier. Ce théorème fait partie d'un ensemble de théorèmes en analyse combinatoire, chacun ayant une adpatation en théorie des graphes.