Graph Exercises 2

Gilles
gilles.richard@irit.fr

1 Objectifs

On veut manipuler des graphes et faire tourner 'a la main' des algorithmes vus en cours.

2 Exemple 1

On considere le graphe $G = (X, U, \omega)$ **non oriente pondere** ou:

- $X = \{x_1, x_2, x_3, x_4, x_5\}$
- $U = \{(x_1, x_2), (x_1, x_3), (x_1, x_4), (x_1, x_5), (x_2, x_3), (x_2, x_4), (x_2, x_5), (x_3, x_4)\}$
- $\omega((x_1, x_2)) = 6, \omega((x_1, x_3)) = 5, \omega((x_1, x_4)) = 1, \omega((x_1, x_5)) = 10,$
  $\omega((x_2, x_3)) = 4, \omega((x_2, x_4)) = 2, \omega((x_2, x_5)) = 8, \omega((x_3, x_4)) = 7$

1. Calculer la densite $d(G)$de ce graphe G.

   SOL.: On utilise la formule pour les graphes non orientes: $\frac{2\times|U|}{|X|(|X|-1)}$ et on obtient $\frac{4}{5}$.

   Quelles sont les aretes manquantes pour avoir une densite de 1?

   SOL: Il manque 2 aretes (x_3, x_5) et (x_4, x_5) pour avoir un graphe complet de densite 1. Dans ce cas, tous les sommets sont relies par une arete a tous les autres.

2. Construire la matrice d'adjacence pour ce graphe.

   SOL.: C'est une matrice 5x5 facile a construire. Matrice symetrique car graphe non oriente.

3. Indiquer pourquoi ce graphe admet un MST.

   SOL.: Car il est connexe. De plus, comme tous les poids sont distincts, ce MST est unique.

4. Appliquer l'algorithme de Prim qui construit un MST. On representera chaque etape k (allant de 0 a $|X| - 1$) de l'algorithme par une ligne de la table 1 Cette table permet de dessiner l'arbre ensuite. Donner la representation

k	*arete selectionnee*	X_k	U_k
0	*aucune*	$\{x_1\}$	$\emptyset$
1	...	...	...

Table 1: Prim

$T = (X, U', \omega)$ de cet arbre. Calculer $\omega(T)$. Calculer sa densite $d(T)$. Donner une formule generale pour la densite d'un arbre couvrant (spanning tree) (minimum ou pas).

SOL.: Le graphe initial G est represente par la figure 1.

La table:

k	*arete selectionnee*	X_k	U_k
0	*aucune*	$\{x_1\}$	$\emptyset$
1	$(x_1, x_4)(1)$	$\{x_1, x_4\}$	$\{(x_1, x_4)\}$
2	$(x_2, x_4)(2)$	$\{x_1, x_2, x_4\}$	$\{(x_1, x_4), (x_2, x_4)\}$
3	$(x_2, x_3)(4)$	$\{x_1, x_2, x_3, x_4\}$	$\{(x_1, x_4), (x_2, x_4), (x_2, x_3)\}$
4	$(x_2, x_5)(8)$	X	$\{(x_1, x_4), (x_2, x_4), (x_2, x_3), (x_2, x_5)\}$

Table 2: Prim

$T = (X, \{(x_1, x_4), (x_2, x_4), (x_2, x_3), (x_2, x_5)\})$ et $\omega(T) = 15$ (voir figure 2). Densite d'un arbre est $\frac{2\times(|X|-1)}{|X|(|X|-1)} = \frac{2}{|X|}$. Dans le cas de notre graphe qui admet 5 sommets, $d(T) = \frac{2}{5}$.

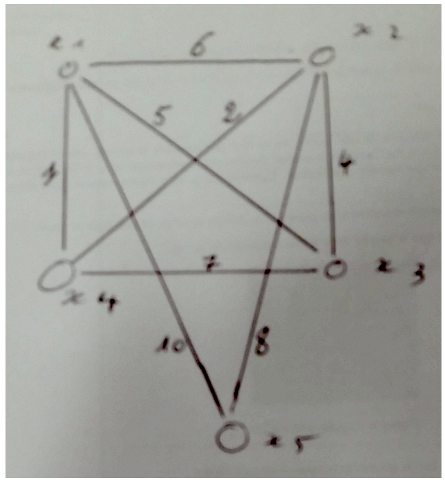

Figure 1: Representation graphique de G

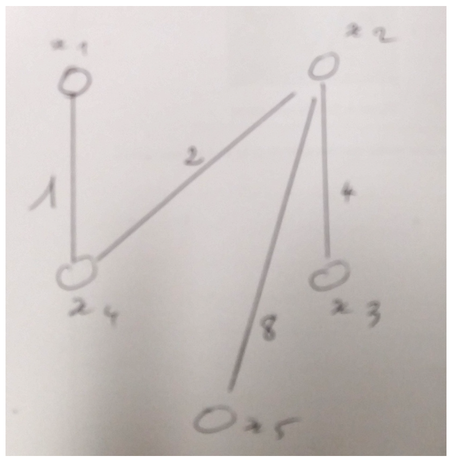

Figure 2: MST pour G

5. Peut il y avoir plusieurs MST pour ce graphe G? Justifier votre reponse.

   SOL.: non car poids tous differents.

6. Indiquer la liste ordonnee des aretes selon ordre croissant des poids. Appliquer l'algorithme de Kruskal elementaire. On representera les etapes de l'algorithme selon la table 3.

k	*arete selectionnee*	U_k	X
0	*aucune*	$\emptyset$	X
..	..	..	..

Table 3: Kruskal basique

   SOL.: On doit trouver un graphe avec 4 aretes puisque le graphe possede 5 sommets. La condition d'arret de l'algorithme est justement d'avoir ajoute 4 aretes. Liste ordonnee des aretes selon poids croissant:

$$(x_1, x_4)(1), (x_2, x_4)(2), (x_2, x_3)(4), (x_1, x_3)(5), (x_1, x_2)(6), (x_3, x_4)(7), (x_2, x_5)(8), (x_1, x_5)(10)$$

   Remarquons qu'a l'etape 4, l'arete la plus courte restante est (x_1, x_3) mais l'ajouter aux 3 autres aretes introduirait un cycle, donc on doit passer a (x_1, x_2) (meme probleme) puis (x_3, x_4) (meme probleme) puis la solution qui est (x_2, x_5). On peut dessiner l'arbre et son poids est $\omega(T) = 1 + 2 + 4 + 8 = 15$. Comme les poids sont tous distincts, le MST est unique et on retrouve le MST calcule avec l'algorithme de Prim.

7. On implemente maintenant le graphe G avec une structure union-find que l'on represente par la table 5 a 2 lignes et $|X|$ colonnes. La deuxieme ligne est censee representer les noeuds pere des sommets correspondant a la premiere ligne.

k	*arete selectionnee*	U_k	X
0	*aucune*	$\emptyset$	X
1	(x_1, x_4)	$\{(x_1, x_4)\}$	X
2	(x_2, x_4)	$\{(x_1, x_4), (x_2, x_4)\}$	X
3	(x_2, x_3)	$\{(x_1, x_4), (x_2, x_4), (x_2, x_3)\}$	X
4	(x_2, x_5)	$\{(x_1, x_4), (x_2, x_4), (x_2, x_3), (x_2, x_5)\}$	X

Table 4: Kruskal basique solution

etape	x_1	x_2	x_3	x_4	x_5	*arete*
0	x_1	x_2	x_3	x_4	x_5	$\emptyset$
1	...	...	...	...	...	...

Table 5: Kruskal avec union-find

L'evolution de la deuxieme ligne de cette table permet de visualiser les etapes de l'algorithme de Kruskal avec cette implementation. La table initiale (etape 0) est donc:

Completer la table 5 en executant l'algorithme de Kruskal avec union-find (sans compression).

SOL.:

etape	x_1	x_2	x_3	x_4	x_5	*arete*
0	x_1	x_2	x_3	x_4	x_5	$\emptyset$
1	x_4	x_2	x_3	x_4	x_5	(x_1, x_4) YES $CC(x_1) = x_1$ et $CC(x_4) = x_4$: fusion(x_1, x_4)
2	x_4	x_4	x_3	x_4	x_5	(x_2, x_4) YES $CC(x_2) = x_2$ et $CC(x_4) = x_4$: fusion(x_2, x_4)
3	x_4	x_4	x_4	x_4	x_5	(x_2, x_3) YES $CC(x_2) = x_4$ et $CC(x_3) = x_3$: fusion(x_3, x_4)
4 NO	x_4	x_4	x_4	x_4	x_5	(x_1, x_3) NO $CC(x_1) = CC(x_3) = x_4$
4 NO	x_4	x_4	x_4	x_4	x_5	(x_1, x_2) NO $CC(x_1) = CC(x_2) = x_4$
4 NO	x_4	x_4	x_4	x_4	x_5	(x_3, x_4) NO $CC(x_3) = CC(x_4) = x_4$
4	x_4	x_4	x_4	x_4	x_4	(x_2, x_5) YES $CC(x_2) = x_4$ et $CC(x_5) = x_5$: fusion(x_5, x_4)

Table 6: Kruskal avec union-find

A la fin tout le monde est dans la meme composante connexe (heureusement) MAIS le MST est obtenu en regardant les aretes que nous avons ajoutees. Et on retombe sur le MST de Prim. Pour $fusion(x, y)$, on a toujours fait $p[x] = y$.

3 Exemple 2

On execute l'algorithme de Kruskal sur le graphe G de la figure 3, en utilisant la structure union-find. A une certaine etape de l'algorithme, la table represente l'arborescence de la figure 4.

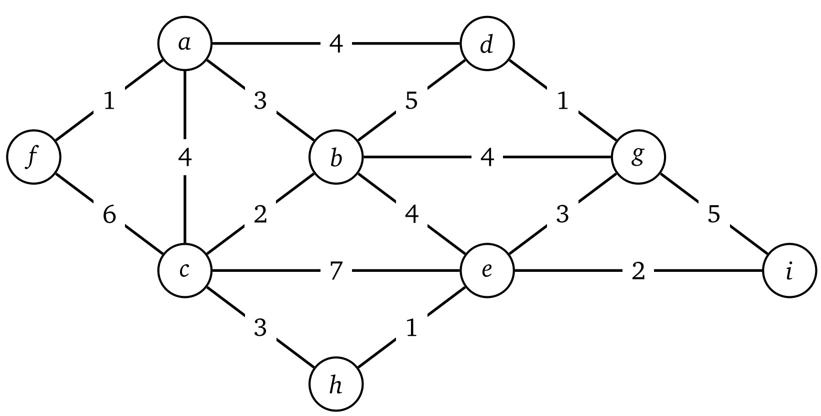

Figure 3: Graphe 1

En fonction de l'arborescence:

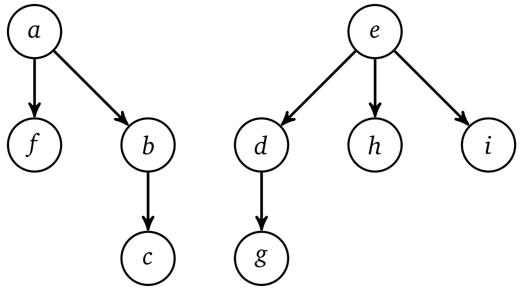

Figure 4: Arborescence courante

1. Indiquer le nombre d'aretes du graphe initial qui ont ete utilisees.

   SOL.: 7 car il y a 7 fleches, chacune consiste en l'utilisation d'une arete du graphe initial.

2. Faire la liste ordonnee par poids croissant des aretes du graphe G.

   SOL.: $\{af, dg, he[1], ei, cb[2], ab, eg, ch[3], ac, ad, be, bg[4], bd, gi[5], fc[6], ce[7]\}$

3. En deduire la liste des aretes qui ont deja ete rajoutees.

   SOL.: Les 7 aretes utilisees sont $af, he, dg, ei, cb, ab, eg$. L'arete ch n'a pas encore ete utilisee puisque c et h ne sont pas dans la meme composante connexe a ce stade de l'algorithme.

   Note: On rappelle qu'une arete de l'arborescence comme (e, d), signifie simplement qu'il existe une chemin de e a d. Le nombre d'arcs de l'arborescence est egal au nombre d'aretes (parmi les aretes du graphe initial) que l'on a deja ajoutees dans la construction a la Kruskal.

4. Completer l'algorithme pour obtenir le MST. Donner le MST sous la forme $T = (X, U)$.

   SOL.: Il faut rajouter ch qui, comme on le voit dans l'arborescence, ne sont pas dans la meme composante connexe. A ce stade, on a donc exactement 8 aretes c'est a dire le nombre voulu pour s'arreter car on a 9 sommets.. Il ne reste qu'a representer par un schema l'arbre obtenu en se basant sur les aretes retenues $T = (X, \{af, he, dg, ei, cb, ab, eg, ch\})$ et $\omega(T) = 16$. Voir schema sur figure 5.

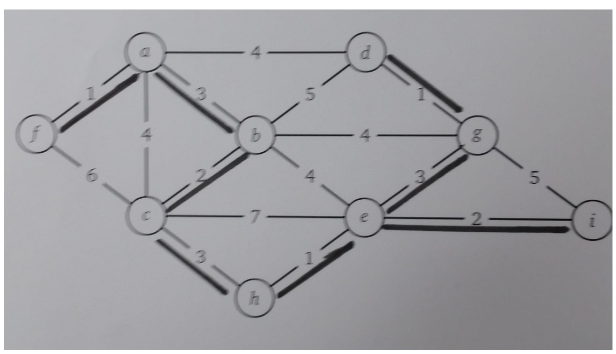

Figure 5: MST pour G

4 Exemple 3

On s'interesse ici a des graphes orientes et a la notion de circuit. L'absence de circuit est indispensable a l'utilisation d'un grand nombre d'algorithmes en theorie des graphes. Il est donc interessant de considerer le probleme de la *detection de circuit*.

1. On considere le graphe G oriente de la figure 6. Construire la matrice d'adjacence M de ce graphe.

SOL.:

$$\begin{array}{c|cccc} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & 0 & 1 & 0 & 1 \\ x_2 & 0 & 0 & 1 & 1 \\ x_3 & 1 & 0 & 0 & 1 \\ x_4 & 0 & 0 & 0 & 0 \end{array}$$

2. Construire la fermeture transitive $\hat{G}$ du graphe G. Rappelons que $\hat{G} = (X, \hat{U})$ ou $(x, y) \in \hat{U}$ si et seulement si il existe une chemin de x vers y.

   SOL.: On rajoute les arcs $(1, 3), (2, 1), (3, 2)$ qui sont les seules aretes manquantes pour rendre le graphe transitif. On peut donc facilement construire la matrice d'adjacence de ce nouveau graphe.

$$\begin{array}{c|cccc} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & 0 & 1 & 1 & 1 \\ x_2 & 1 & 0 & 1 & 1 \\ x_3 & 1 & 1 & 0 & 1 \\ x_4 & 0 & 0 & 0 & 0 \end{array}$$

3. Calculer M^2 et M^3. Est ce que M^2 et M^3 sont toujours des matrices d'adjacence (pourquoi?)

   SOL.: M^2: On verifie que 1 apparait cellule (i, j) la ou il existe un chemin de longueur 2 pour aller de x_i a x_j.

$$\begin{array}{c|cccc} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & 0 & 0 & 1 & 1 \\ x_2 & 1 & 0 & 0 & 1 \\ x_3 & 0 & 1 & 0 & 1 \\ x_4 & 0 & 0 & 0 & 0 \end{array}$$

   M^3: On verifie que 1 apparait cellule (i, j) la ou il existe un chemin de longueur 3 pour aller de x_i a x_j.

$$\begin{array}{c|cccc} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & 1 & 0 & 0 & 1 \\ x_2 & 0 & 1 & 0 & 1 \\ x_3 & 0 & 0 & 1 & 1 \\ x_4 & 0 & 0 & 0 & 0 \end{array}$$

   Les matrices obtenues ne sont pas toujours des matrices d'adjacence car elles ne comportent pas necessairement que des 0 ou des 1. Par exemple, $M^2[x_i, x_j]$ est le nombre de chemins de longeur exactement 2 entre x_i et x_j. S'il n'y a qu'un seul chemin, alors on obtient 1 mais s'il y en a plusieurs, on obtient un nombre plus grand que 1. Par exemple, si on considere le graphe fermeture transitive du graphe initial G, on voit qu'il y a 2 chemins de longueur 2 pour aller de 1 a 4: le chemin $1 \to 2 \to 4$ et $1 \to 3 \to 4$: donc, la valeur correspondante de la matrice d'adjacence est 2.

4. Si on remplace tous les elements non nuls de la matrice M^2 par des 1, on obtient donc une matrice d'adjacence M'^2. Quel est le graphe correspondant?

   SOL.: Le graphe correspondant est (X, U^2) ou U^2 est constitue des arcs (x_i, x_j) tels que il existe un chemin de longueur exactement 2 entre x_i et x_j dans le graphe G initial.

5. Meme question pour M^3 (on obtient M'^3).

   SOL.: Le graphe correspondant est (X, U^3) ou U^3 est constitue des arcs (x_i, x_j) tels que il existe un chemin de longueur exactement 3 entre x_i et x_j dans le graphe G initial.

6. Considerons maintenant la somme de matrices $M + M'^2 + M'^3$, ou au lieu de considerer la somme usuelle des reels, on considere la somme Booleenne ou $1 + 1 = 1$ (on considere la somme Booleenne comme interpretant le OU logique et non pas comme le XOR). On obtient donc une matrice d'adjacence M^T. Construire le graphe correspondant. Que remarquez vous?

   SOL.: D'une maniere generale, si on note M^T la matrice ainsi obtenue en ajoutant les puissances de la matrice initiale, $M^T[x_i, x_j] = 1$ si et seulement si il existe un chemin de x_i vers x_j dans le graphe G initial (on n'a pas d'info sur sa longueur mais on s'en fiche pour l'instant.). Cette matrice est donc bien la matrice d'adjacence de la fermeture transitive du graphe initial.

7. En deduire une relation entre la fermeture transitive d'un graphe G et l'existence d'un circuit dans ce graphe G.

   SOL.: Puisque $M^T[x_i, x_i] = 1$ ssi il y a un chemin entre x_i et x_i dans le graphe G initial, c'est a dire un circuit, verifier que la diagonale de la matrice ne contient pas de 1 permet d'affirmer que le graphe est sans circuit, et avec circuit dans le cas contraire. Donc pour tester circuit ou pas, faire matrice d'ajdacence M, calculer matrice d'adjacence de la fermeture transitive M^T, regarder la diagonale de cette matrice M^T.

8. Appliquer cette methode pour verifier que le graphe de la figure 6 possede un circuit.

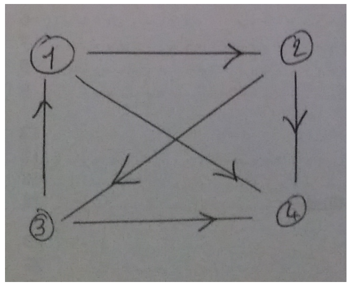

Figure 6: Graphe avec circuit

   SOL.: Deja avec M^3, on a des 1 sur la diagonale... Ces 1 resteront sur la diagonale de la fermeture transitive.

9. Appliquer cette methode pour verifier que le graphe de la figure 7 est sans circuit.

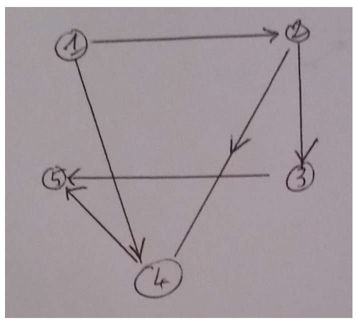

Figure 7: Graphe sans circuit

   SOL.: A vous de jouer, rien de terrible mais un peu...boring.

10. Appliquer la methode des niveaux pour verifier que le graphe de la figure 7 est sans circuit.

    SOL.:

$$N_0 = \{1\}, N_1 = \{2\}, N_3 = \{3, 4\}, N_4 = \{5\}$$

$X = N_0 \cup \ldots \cup N_4$ donc G est sans circuit. De plus on a un tri topologique $\{1, 2, 3, 4, 5\}$ (ou $\{1, 2, 4, 3, 5\}$).