Graph Exercises 3

Gilles
gilles.richard@irit.fr

1 Introduction

On travaille sur matrices adjacence, matrices d'incidence, les MST et parcours d'arbres.

2 Exercice 1

On donne le graphe $G = (X, U)$ **non oriente** suivant de la figure 1:

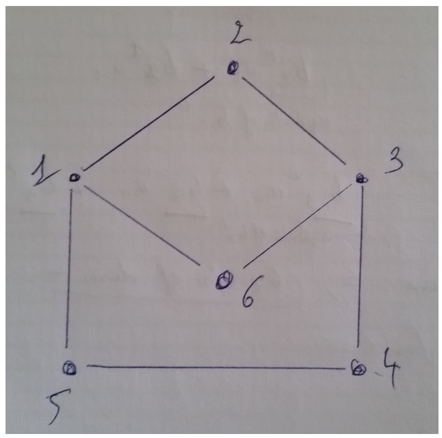

Figure 1: Representation graphique de G

1. Calculer son diametre et sa densite.

   SOL.: On a $|X| = 6$ et $|U| = 7$, d'ou la densite pour graphe non oriente:

   $$d(G) = \frac{2|U|}{|X| \times (|X| - 1)} = \frac{14}{30} = \frac{7}{15} \equiv \frac{1}{2}$$

   Le diametre est 2 puisque on peut aller de tout sommet a tout autre par un chemin de longueur 2 maximum. Le graphe est evidemment connexe.

2. Construire sa matrice d'adjacence.

   SOL.: La matrice est symetrique puis le graphe est non oriente. C'est une matrice 6×6: La notion de matrice

	1	2	3	4	5	6
1	0	1	0	0	1	1
2	1	0	1	0	0	0
3	0	1	0	1	0	1
4	0	0	1	0	1	0
5	1	0	0	1	0	0
6	1	0	1	0	0	0

Table 1: Matrice d'adjacence de G

d'incidence n'a pas de sens pour graphe non oriente car il n'y a pas de notion d'origine et d'extremite.

3. Contruire les listes d'adjacence. Verifier la relation entre $|U|$ et les longueurs des listes d'adjacence.

   SOL.: L'ordre des fils d'un sommet n'a pas d'importance.
   $1 \rightarrow 2 \rightarrow 5 \rightarrow 6$
   $2 \rightarrow 1 \rightarrow 3$
   $3 \rightarrow 2 \rightarrow 4 \rightarrow 6$
   $4 \rightarrow 3 \rightarrow 5$
   $5 \rightarrow 1 \rightarrow 4$
   $6 \rightarrow 1 \rightarrow 3$
   En additionnant les longueurs des listes d'adjacence, on obtient bien $14 = 2 \times |U|$.

4. Le graphe G est-t-il transitif (autrement dit la relation definie par U est-t-elle une relation transitive)? Si non, combien doit on ajouter d'aretes pour le rendre transitif?

   SOL.: NON puisque l'on a bien un un chemin de 1 a 3 mais on n'a pas d'arete entre 1 et 3. On s'apercoit que tout sommet y est accessible a partir de tout sommet x avec un chemin de longueur 2 maximum: donc pour tous les sommets (x, y) tels qu'il existe un chemin de x a y, nous devons ajouter l'arete (x, y) si elle n'existe pas. En d'autres termes, on va rendre le graphe complet en le rendant transitif car on aura toutes les aretes possibles (x, y). On aura donc $\frac{6\times5}{2} = 15$ aretes. On doit donc rajouter $15 - 7 = 8$ aretes au graphe G.

5. Y a t il des cliques dans le graphe G? Rajouter une arete pour construire une clique de cardinal 3.

   SOL.: Oui, toute paire (x, y) telle que $(x, y) \in U$ constitue une clique. Il n'y a aucune clique plus importante que les autres en terme de taille. Si on rajoute l'arete $(5, 6)$, alors $(1, 5, 6)$ constitue aussi une clique.

6. Faire un parcours en largeur du graphe G en partant du sommet 6. Puis un parcours en profondeur

   SOL.: En partant de 6, on parcourt les sommets a distance 1, puis 2, etc.
   distance 1: $1 \rightarrow 3$
   distance 2: $5 \rightarrow 2 \rightarrow 4$
   Parcours en profondeur en partant de 6
   $6 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 4$ par exemple.

7. Maintenant on ajoute une fonction de poids ω au graphe G que l'on schematise sur la figure 2:

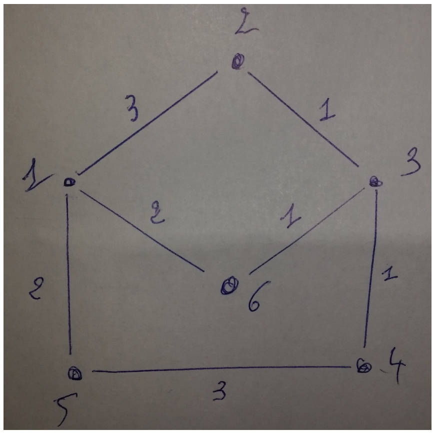

Figure 2: G pondere

8. Contruire un MST d'abord en utilisant l'algo de Prim. On considere que le sommet de depart est 6. Indiquer son poids.

   SOL.: G est connexe donc il admet (au moins) un MST T. Prim d'abord. On ecrit une table comme vue en cours.

k	*arete selectionnee*	X_k	U_k
0	*aucune*	$\{6\}$	$\emptyset$
1	$(6,3)(1)$	$\{6,3\}$	$\{(6,3)\}$
2	$(3,2)(1)$	$\{2,3,6\}$	$\{(6,3),(3,2)\}$
3	$(3,4)(1)$	$\{2,3,4,6\}$	$\{(6,3),(3,2),(3,4)\}$
4	$(6,1)(2)$	$\{1,2,3,4,6\}$	$\{(6,3),(3,2),(3,4),(6,1)\}$
5	$(1,5)(2)$	$\{1,2,3,4,5,6\}$	$\{(6,3),(3,2),(3,4),(6,1),(1,5)\}$

Table 2: Prim

Le poids $\omega(T) = 7$.

9. En utilisant Kruskal (basique), donner la liste des aretes constituant un MST T pour G.

   SOL.: On peut ecrire une liste (non unique) ordonnee par poids croissant des aretes et on trouve :

   $$\{(6,3),(3,4),(3,2),(1,5),(1,6),(5,4),(1,2)\}$$

   En fait, l'ajout des 5 premieres aretes dans l'ordre des poids croissants ne genere pas de cycle.

   $$\{(6,3),(3,4),(3,2),(1,5),(1,6)\}$$

   Donc, il n'y a qu'un seul MST malgre le fait que les poids des aretes ne soient pas unique.

10. Appliquer Kruskal avec union-find. On representera l'execution de l'algorithme avec la table habituelle.

    SOL.: Pas de difficulte compte tenu du fait que les 5 premieres aretes ajoutees ne cree pas de cycle. Seule, la deuxieme ligne de la table evolue.

etape	1	2	3	4	5	6	*arete*
0	1	2	3	4	5	6	$\emptyset$
1	1	2	**6**	4	5	**6**	$\{(6,3)\}$
2	1	2	**6**	**6**	5	6	$\{(6,3),(3,4)\}$
3	1	**6**	**6**	**6**	5	6	$\{(6,3),(3,4),(3,2)\}$
4	**5**	6	6	6	**5**	6	$\{(6,3),(3,4),(3,2),(1,5)\}$
5	**6**	6	6	6	5	**6**	$\{(6,3),(3,4),(3,2),(1,5),(1,6)\}$

Table 3: Kruskal avec union-find

En conclusion, exercez vous a repondre aux memes questions mais sur des graphes ayant plus de sommets et d'aretes.

3 Exercice 2

On reflechit un peu a un nouvel algo pour calculer un MST. On considere un graphe $G = (X, U)$ non oriente connexe et pondere (il admet donc au moins un MST).

1. Montrer que toute arete e qui n'est pas dans un cycle de G appartient a tout MST.

   SOL.: Si $e = (x, y)$ n'est pas dans un cycle, cela signifie que le seul chemin dans G menant de x a y est reduit a cette arete e. Supprimer cette arete rend donc le graphe non connexe. Puisque qu'un MST est connexe, il contient necessairement e.

2. Considerons maintenant un cycle C dans ce graphe G et soit e l'arete la plus lourde de C. Montrer qu'il existe un MST pour G qui ne contient pas e.

   SOL.: On peut raisonner par l'absurde en supposant que cette propriete n'est pas vraie et donc qu'il existe un arete $e = (a, b)$ de C, de poids maximum dans C et que tous les MST T la contiennent. Soit alors $e' = (c, d)$ l'arete la plus legere de C. Faire un dessin aide a comprendre. Considerons alors un MST T qui par hypothese contient e et donc un chemin de a a b. Si, dans T, on remplace e par e', on garde la connexite car il y a toujours un chemin de a a b (mais pas le meme). On obtient encore un arbre T' car on a la connexite et le meme nombre d'aretes. Mais $\omega(T') \leq \omega(T)$ donc T' est un MST qui ne contient pas e: contradiction.

3. En deduire un algorithme de calcul de MST.

   SOL.: On ordonne la liste des aretes par poids decroissant:

```
L = sort(U , w, decreasing)
for e in L  //on parcourt L
     if e is part of a cycle in G=(X, L)
        L = L - e
```

   En effet, si e est dans un cycle de $G = (X, L)$, alors d'apres la propriete precedente, il existe un MST qui ne le contient pas et on construit justement ce MST. Et s'il n'est pas dans un cycle, necessairement tout MST contient e. Contrairement a Kruskal qui ajoute des aretes a partir de $\emptyset$, cet algo enleve des aretes a partir de U.

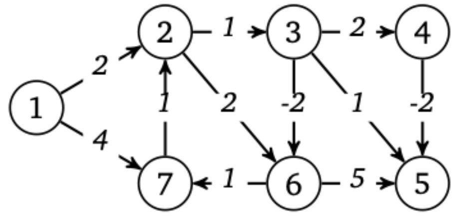

Figure 3: G oriente pondere

4 Exercice 3

On considere le graphe oriente pondere G suivant:

1. G est il connexe quand on le considere comme non oriente?

   SOL.: Oui il est connexe car quelque soit la paire (x, y), il y a un chemin de x a y.

2. G est-il fortement connexe?

   SOL.: En tant que graphe oriente, il n'est pas fortement connexe. En effet, il existe un chemin pour aller de 1 a 6 par exemple, mais aucun chemin de 6 a 1.

3. Construire le graphe reduit associe a G.

   SOL.: $CC1 = \{1\}, CC2 = \{2, 3, 6, 7\}, CC3 = \{4\}, CC4 = \{5\}$. Le graphe reduit s'en deduit:

   Les arcs sont: $\{(CC1, CC2), (CC2, CC3), (CC2, CC4), ((CC3, CC4)\}$.

4. Peut on faire un tri topologique du graphe G?

   SOL.: Non car il y a des cycles.

5. Peut on faire un tri topologique du graphe reduit? Si oui, faire ce tri.

   SOL.: On peut toujours faire le tri topologique d'un graphe reduit car un tel graphe ne comporte jamais de cycle. On met en niveau le graphe reduit $N_0 = \{CC1\}, N_1 = \{CC2\}, N_2 = \{CC3\}, N_3 = \{CC4\}$. Le tri topologique est donc trivial: $CC1, CC2, CC3, CC4$.

6. Appliquer l'algorithme de Bellman pour chercher les plus courts chemins en partant du sommet 1.

   SOL.: Comme tous les algorithmes de parcours de graphe oriente pondere, on represente leur execution par une table qui evolue avec les etapes de l'algorithme. Concernant precisement Bellman, la derniere ligne de la table est entierement mise a jour a chaque etape. On s'arete quand on a autant d'etapes que de sommets ou bien quand la mise a jour de la derniere ligne ne la change pas. Pour mettre a jour la colonne i, on regarde tous les **peres** du sommet i que je note par exemple a, b, c. Et ensuite on considere 3 nombres: $dist(1, a) + w(a, i), dist(1, b) + w(b, i), dist(1, c) + w(c, i)$; Les 3 valeurs $dist(1, a), dist(1, b), dist(1, c)$ sont justement dans la derniere ligne de la table: par exemple, $d(1, a)$ c'est la valeur de la colonne a. On a donc toutes les infos pour calculer ces 3 nombres. On met dans la nouvelle colonne i le plus petit de ces 3 nombres.

etape	1	2	3	4	5	6	7
0	0	∞	∞	∞	∞	∞	∞
1	0	2	∞	∞	∞	∞	4
2	0	2	3	∞	∞	4	4
3	0	2	3	5	4	1	4
4	0	2	3	5	3	1	2
5	0	2	3	5	3	1	2

   Table 4: Bellman en partant du sommet 1

   L'etape 5 ne change pas la derniere ligne de la table donc on s'arete. Comme le nomb re d'etape est egal au nombre de sommets, la derniere ligne indique les distances minimales au sommet 1.

7. Idem mais en partant du sommet 7.

   SOL.:

On voit qu'il n'y a pas de chemin pour aller de 7 a 1 (distance infinie). D'autre part, on peut aller de 7 a 6 en passant par 3 avec un chemin de 3 arcs mais de poids nul!

etape	1	2	3	4	5	6	7
0	∞	∞	∞	∞	∞	∞	0
1	∞	1	∞	∞	∞	∞	0
2	∞	1	2	∞	∞	3	0
3	∞	1	2	4	3	0	0
4	∞	1	2	4	2	0	0
5	∞	1	2	4	2	0	0

Table 5: Bellman en partant du sommet 7