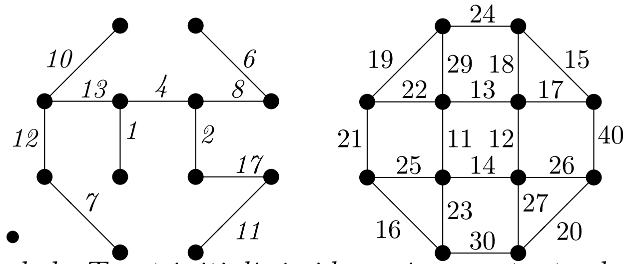
Graphes, TD N^o 3 : Arbres Couvrants Minimaux

- *petit rappel memo :*
  algo ou on part d'un point et ou on propage = Prim
  algo ou on travaille partout a la fois = kruskal
  une seance pour Q1 Q2 Q3a Q3b

1

Faire tourner l'algorithme de Kruskal et celui de Prim (partir du sommet en dessous de l'arête 40) sur le graphe suivant :



Kruskal : T est initialisé vide, puis pour toutes les arêtes a prises par ordre croissant de poids, si $T + a$ n'a pas de cycle alors on ajoute a à T.
Si on nomme les arêtes par leur poids, l'algorithme donne ça : on prend 11, on prend 12, on prend 13, on rejette 14, on prend 15, on prend 16, on prend 17, on rejette 18, on prend 19, on prend 20, on prend 21, on prend 22, on rejette 23, on rejette 24, on rejette 25, on prend 26, et c'est fini.
On note qu'on peut savoir facilement quand il faut s'arrêter car le nombre d'arêtes ajoutées a atteint $n - 1$.

Prim : on part de x_0, et on ajoute les arêtes dans cet ordre :
20, 26, 12, 13, 11, 17, 15, 22, 19, 21, 16.
À chaque étape, les arêtes possibles sont :
$\{20, 26, 40\}$
$\{26, 27, 30, 40\}$
$\{12, 14, 30, 40\}$
$\{13, 14, 17, 18, 30, 40\}$
$\{11, 14, 17, 18, 22, 29, 30, 40\}$
$\{17, 18, 22, 23, 25, 29, 30, 40\}$
$\{15, 18, 22, 23, 25, 29, 30\}$
$\{22, 23, 24, 25, 29, 30\}$
$\{19, 21, 23, 24, 25, 29, 30\}$
$\{21, 23, 25, 30\}$
$\{16, 23, 30\}$

2

Pour chaque question, donnez une preuve directe et une autre qui utilise les propriétés vues en cours des algorithmes de Prim et de Kruskal.

- *Prop vue en cours : Prim et Kruskal construisent QUE des ACM et TOUS les ACM. IE que toute exécution donne un ACM, et pour tout ACM, il y a une exécution qui donne cet ACM*

— Montrez que s'il existe une seule arête de poids minimal, alors elle appartient à tous les arbres couvrants minimaux du graphe.

- *On le montre par l'absurde : soit T un ACM de G, supposons que T ne contient pas a. Alors $T + a$ contient un cycle, et si on enlève n'importe quelle arête b de ce cycle, $T + a - b$ sera de nouveau un arbre couvrant de G (car il est connexe à n sommets et possède $n - 1$ arêtes).*
  Or si on prend $b \neq a$ dans le cycle, on a $w(b) > w(a)$ puisque aa est l'unique arête de poids minimal, donc $w(T + a - b) = w(T) - w(b) + w(a) < w(T)$. Donc $T + a - b$ est un arbre couvrant de poids strictement inférieur à T, donc T n'est pas un arbre couvrant minimal.
  Donc T est un ACM $\Rightarrow$ a est dans T.

  Soit T un ACM, il existe une exécution de Kruskal qui donne T (prop TOUS). Or la première étape de cette exécution est forcément de choisir a et de la prendre, puisqu'elle ne peut pas faire de cycle toute seule. Donc T contient a.

— Montrez que si a est une arête minimale, alors il existe un ACM qui la contient.

- *Soit T un ACM. Soit T contient a, auquel cas c'est bon, soit $T + a$ contient un cycle. Alors soit b une arête du cycle autre que a, $T + a - b$ contient a et est un arbre couvrant, et $w(T + a - b) = w(T) - (w(b) - w(a)) \leqslant w(T)$. Donc comme $w(T)$ est minimal, on a bien $w(T + a - b)$ aussi minimal et on a trouvé un ACM qui contient a.*

  Comme a est de poids minimal, on peut lancer Kruskal en choisissant a en premier, et l'algorithme prendre cette arête puisqu'elle ne ferme pas de cycle ; on obtient alors un ACM (prop QUE) qui contient a.

— Soit a une arête de poids minimal. Montrez qu'il existe un ACM T ne contenant pas a ssi il existe un cycle ne contenant que des arêtes de poids minimal et qui contient a.

- *Soit T un ACM, supposons que T ne contient pas a. Alors $T + a$ contient un cycle. Soit b une arête de ce cycle, on a montré que $T + a - b$ est également un ACM, donc $w(T) = w(T) + w(a) - w(b)$, donc $w(a) = w(b)$. Donc toutes les arêtes du cycle sont de même poids minimal.*

  Si on lance Kruskal sur le graphe, il va commencer par choisir toutes les arêtes de poids minimal. Il les prendra donc toutes sauf si quand il arrive à l'arête a, elle et les autres déjà choisies (donc toutes de poids minimal) forment un cycle. Donc tous les ACM contiennent toutes les arêtes de poids minimal (prop TOUS) sauf si il existe un cycle formé uniquement d'arêtes de poids minimal.

3

Un graphe $G = (S, A)$ et une pondération w sont donnés. Soit a une arête de G, on note G_a le graphe G privé de l'arête a. i.e. $G_a = (S, A - \{a\})$. La pondération est conservée pour G_a. On suppose que G_a (et donc a forciori G) est connexe.

— On dispose de T_a un ACM pour G_a. Comment en déduire un ACM T pour G ?

- *On ajoute a à T_a : le graphe obtenu contient exactement un cycle C_a. Alors soit a est de poids maximal, et on garde T_a, soit il existe b de poids maximal dans C_a tq $w(b) > w(a)$, et dans ce cas on choisit $T_a + a - b$.*
  Prouvons que l'arbre obtenu est bien un ACM de G. Pour ça, on va utiliser un lemme : T un arbre inclu dans G est un ACM de G ssi pour toute arête c hors de T, toutes les arêtes du cycle formé par l'ajout de c à T ont un poids inférieur ou égal à celui de c.
  Preuve du lemme :

— *si il existe c tq le cycle de $T+c$ contient une arête a de poids $w(a) > w(c)$, alors $T+c-a$ est un arbre couvrant de G, puisqu'il a $n-1$ arêtes et qu'il est acyclique, et il est de poids $w(T)+w(c)-w(a) < w(T)$: alors T n'est pas un ACM;*

— *si T n'est pas un arbre couvrant minimal, on va considérer une exécution de Kruskal qui favorise les arêtes de T (c'est-à-dire qu'elle considère toutes les arêtes de T d'un certain poids avant de considérer celles du même poids hors de T). Comme T n'est pas un ACM, au bout d'un moment l'algorithme va ajouter une arête hors de T ; on note a la première telle arête. Comme a est ajoutée par l'algorithme, on sait qu'elle ne forme pas de cycle avec les arêtes déjà ajoutée par l'algorithme.*

  On note également que comme avant a on n'a ajouté que des arêtes de T, aucune arête de T considérée n'a pu encore être refusée, car il n'existe pas de cycle formé seulement d'arêtes de T.

  Donc, si on considère le cycle de $T+a$, au moins une de ses arêtes est considérée par l'algorithme après a, puisque toutes les autres sont dans T et qu'elles n'ont pas toutes été ajoutées, ou a aurait été refusé. Or, comme les arêtes de T sont favorisées par l'algorithme, ça veut dire que cette arête considérée après a est de poids strictement supérieur à celui de a. Donc a n'est pas de poids maximal dans le cycle de $T+a$.

  Ainsi, si T est un arbre couvrant non minimal, il existe une arête a qui n'est pas de poids maximal dans le cycle de $T+a$.

À partir de ce lemme, on étudie l'arbre donné par la méthode précédemment décrite.

Si on a gardé $T = T_a$, alors a n'est pas dans T et la condition du lemme est vérifiée pour a (i.e a est de poids maximal dans l'unique cycle de $T+a$). Elle est également vérifiée pour toute arête c qui est dans $G_a \setminus T_a$, puisque T_a est un ACM de G_a. Donc dans ce cas, la condition est bien vérifiée pour toute arête extérieure à T, et donc T est un ACM.

Dans le deuxième cas, où $T = T_a + a - b$, on considère une arête quelconque c qui n'est pas dans T. On note d'abord que si $c = b$, alors le cycle créé par l'ajout de c est C_a et la condition est bien respectée. Trois cas sont possible :

— *$c = b$: par hypothèse sur b, on a bien b de poids maximal dans C_a, qui est entièrement inclus dans $T+b$, donc qui est l'unique cycle obtenu en ajoutant b à T, donc la condition est bien respectée;*

— *le cycle C_c de $T+c$ ne contient pas a, auquel cas il est entièrement inclus dans T_a+c, et donc c'est l'unique cycle de T_a+c, donc par l'application du lemme à T_a c est bien de poids maximal dans C_c ;*

— *le cycle C_c contient bien a et $c \neq b$. Alors ça veut dire que c n'est pas non plus dans T_a, et le cycle C'_c qui est l'unique cycle de T_a+c passe par b (sinon C'_c serait dans $T+c$ et ne passerait pas par a).*

  Dans ce cas, on remarque que $C'_c \cup C_a - b$ contient un cycle passant par c, puisqu'on peut aller d'une extrémité 1 de c à une extrémité 1 de b par $C'_c - b - c$, puis de l'extrémité 1 de b à l'extrémité 2 de b (l'autre) par $C_a - b$, puis de l'extrémité 2 de b à l'extrémité 2 de c en prenant l'autre morceau de $C'_c - b - c$. On a donc bien un chemin dans $C'_c \cup C_a - b - c$ entre les deux extrémités de c, donc un cycle contenant c.

  Or ce cycle est entièrement contenu dans $T+c$, donc c'est C_c qui est inclus dans $C'_c \cup C_a$.

  Or, comme $b \in C'_c$, on sait que $w(c) \geqslant w(b)$. Donc soit d une arête de $C_c - c$, soit d est dans C'_c et donc comme T_a est un ACM de G_a, par le lemme, $w(d) \leqslant w(c)$, soit $d \in C_a$ et donc par maximalité du poids de b dans C_a, $w(d) \leqslant w(b) \leqslant w(c)$. Donc dans tous les cas $w(c) \geqslant w(d)$, et donc c est de

poids maximal dans C_c.
Donc l'arbre T est bien un ACM.

— On dispose de T un ACM pour G. Comment en déduire un ACM T_a pour G_a ?

- *Si T ne contient pas a, on garde $T_a = T$: le lemme donné dans la question précédente nous dit que la qualité d'ACM est alors conservée.*
  Si T contient a, on trouve un arbre couvrant de G_a en ajoutant l'arête de poids minimal de $G - a$ qui n'ajoute pas de cycle à $T - a$, et donc qui relie les deux composantes connexes de $T - a$.
  Vérifions que dans ce cas on obtient bien un ACM de G_a : on note b l'arête qu'on a ajouté à $T - a$ pour obtenir T_a.
  On remarque que le retrait de a de T a séparé les sommets de G en deux composantes connexes de $T - a$, qu'on notera S_1 et S_2, et qui sont non vide puisqu'il y a une extrémité de a dans chacune. De plus, les arêtes qui n'ajoutent pas de cycle à $T - a$ sont exactement les arêtes ayant une extrémité dans chaque ensemble.
  Considérons une arête c de G_a qui soit hors de T_a. Si les deux extrémités de c sont dans S_1 (resp. dans S_2), alors il existe un chemin entre ces deux extrémités dans $T - a$ qui est contenu entièrement dans S_1 (resp. S_2), et donc le cycle de $T_a + c$ est le même que celui de $T + c$, donc comme T est un ACM de G, c est bien de poids maximal dans le cycle de $T_a + c$.
  Si les extrémités de c sont chacune dans une composante, alors le cycle de $T + c$ contient a et celui de $T_a + c$ contient b. On note que comme c est entre S_1 et S_2, comme b a été choisie de poids minimal, $w(b) \leqslant w(c)$. Considérons C le cycle de $T + c$ et C' celui de $T + b$, on remarque que pour toute arête dans C, son poids est majoré par celui de c et pour toute arête dans C' son poids est majoré par celui de b, donc aussi par celui de c.
  On remarque enfin que comme il existe un chemin dans $C \cup C' - a - c$ entre les deux extrémités de c (entre l'extrémité 1 de c et l'extrémité 1 de a dans C puis entre les deux extrémités de a dans C' puis entre l'extrémité 2 de a et l'extrémité 2 de c dans C), et que $C \cup C' - a - c \subset T_a$, le cycle de $T_a + c$ est inclus dans $C \cup C' - a$, donc les poids des arêtes sont majorés par celui de c.
  Donc la propriété du lemme est bien vérifiée pour toutes les arêtes de G_a hors de T_a, et donc T_a est bien un arbre couvrant minimal de G_a.

4

S'il y a unicité de l'arbre couvrant T minimal, peut-il y avoir deux arêtes de même poids ? En particulier, peut-il y avoir deux arêtes de même poids (1) qui sont toutes les deux dans T ? (2) dont aucune n'est dans T ? (3) dont une seule est dans T ?

- *La réponse est oui à toutes ces question.*
  Exemple 1 : un triangle dont les arêtes sont de poids 1, 1, 2.
  Exemple 2 : un carré avec ses diagonales dont les poids sont 1, 2 3, 4 sur le carré et 5, 5 sur les diagonales.
  Exemple 3 : un carré avec une diagonale nord-ouest − > sud-est dont les poids sur les arêtes sont 1 en bas, 2 à gauche, 3 à droite et sur la diagonale et 4 en haut.

5

Soit G un graphe non orienté valué connexe, non réduit à un arbre Soit T un arbre couvrant minimal. Soit C un cycle.

(a) On suppose que toutes les valuations de G sont différentes. Montrez que l'arête de poids maximum de C n'est pas dans T.

- *On sait qu'il existe une exécution de l'algorithme de Prim qui produit T. Étudions ce qui se passe si l'arête a de poids maximal dans C est dans l'ensemble des arêtes possibles à une certaine étape.*
  Notons A l'ensemble des sommets de l'arbre partiellement construit (i.e l'ensemble des points déjà adjacents à une arête choisie ou bien le sommet de départ si on vient de commencer). Si a est dans l'ensemble des arêtes possible, ça signifie que ses extrémités sont un sommet u dans A et un sommet v pas encore dans A (sinon a serait soit pas encore adjacente à A, soit refermerait un cycle dans l'arbre partiel déjà construit).
  Numérotons maintenant les sommets de C : $u = u0$, $v = u_1$ et pour tout i, l'arête $u_i u_{i+1}$ existe et fait partie de C. Notons i_0 l'indice du premier sommet de C en partant de v et en suivant la numérotation qui est dans A. On note que ce sommet existe, puisque u_0 est dans A, mais que $i_0 = 0$ ssi il n'y a aucun autre sommet de C qui soit dans A.
  Alors l'arête $u_{i_0-1}u_{i_0}$ (ou $u_{|C|-1}u_0$ si $i_0 = 0$) est dans C et est entre un sommet de A et un sommet pas dans A ; elle est donc dans l'ensemble des arêtes possibles à cette étape de l'algorithme. De plus, comme elle est différente de a, elle est de poids strictement inférieur à a ; a n'est donc pas l'arête possible de plus petit poids à cette étape et elle ne peut donc pas être sélectionnée par l'algorithme.
  Ce raisonnement pouvant être appliqué à toutes les étapes de l'algorithme où a est une arête possible, on en déduit que a n'est jamais choisie et donc que a n'est pas dans T.

(b) On ne suppose plus que toutes les valuations de G sont différentes. Montrez qu'il y a au moins une arête de poids maximum de C qui n'est pas dans T, et que pour toute arête a de poids maximum de C, il y a un ACM $\overline{T}$ qui ne la contient pas.

- *Le raisonnement précédent nous dit que soit a une arête de C, si lors d'une exécution de l'algorithme de Prim a est dans l'ensemble des arêtes possibles, alors au moins une autre arête du cycle y est aussi.*
  Supposons par l'absurde qu'un ACM T contient toutes les arêtes de poids maximal de C. Considérons alors une exécution de l'algorithme de Prim qui donne T, et notons a la dernière arête de poids maximal de C à être ajoutée à T. À l'étape où l'algorithme la choisit, il y a au moins deux arêtes de C dans l'ensemble des arêtes possibles ; la deuxième doit donc être de poids au moins égal à celui de a. Mais ce n'est pas possible, car toutes les arêtes de poids égal à celui de a sont déjà dans T, donc ne sont pas dans l'ensemble des arêtes possibles... donc a ne peut pas être ajoutée à T à cette étape, ni à aucune autre.
  Donc aucun arbre couvrant ne peut contenir toutes les arêtes de poids maximal d'un cycle C.

  De même, si on distingue en début d'exécution de l'algorithme de Prim une arête a de poids maximal, cette même conclusion signifie qu'à chaque étape d'une exécution de l'algorithme, on peut faire un choix autre que a, puisque toutes les arêtes de C sont de poids inférieur ou égal à celui de a. Donc il existe une exécution durant laquelle on ne choisit jamais a, donc il existe un ACM ne contenant pas a.