
TD : Graphes non pondérés

1 Applications du cours

Exercice 1. Montrer que dans un graphe non orienté à $n \geq 2$ sommets, il existe deux sommets ayant même degré.

Exercice 2. *Graphe transposé.* Écrire une fonction permettant d'obtenir la représentation par listes d'adjacence du graphe transposé d'un graphe donné par listes d'adjacence. Même question avec un graphe donné par sa matrice d'adjacence.

Exercice 3. *Puits dans un graphe.* On se donne un graphe orienté à n sommets représenté par sa matrice d'adjacence. La plupart des algorithmes sur des graphes donnés par matrices d'adjacence sont de complexité $\Omega(n^2)$ puisqu'il faut en général parcourir la matrice au moins une fois, une exception notable est celui de l'existence d'un *puits* dans le graphe : un sommet v de degré sortant 0 et de degré entrant $n - 1$. Donner un algorithme de complexité $O(n)$ déterminant si un graphe possède un puits. L'écrire en Caml.

Exercice 4. On se donne un graphe non orienté $G = (V, E)$. Décrire un algorithme en $O(|V| + |E|)$ testant si G est un arbre (au sens des graphes). L'implémenter en Caml, en utilisant si besoin une fonction du cours.

Exercice 5. *Tri topologique : autre algorithme.* Dans le cours, on a montré qu'un graphe orienté sans circuit possède un tri topologique, via un lemme montrant qu'un tel graphe possède un sommet de degré sortant 0. Dédurre de cette preuve une alternative au parcours en profondeur permettant de calculer un tri topologique, et l'implémenter en Caml avec une complexité $O(|V| + |E|)$ pour un graphe $G = (V, E)$ donné par listes d'adjacence. Indication : calculer le tableau des degrés des sommets, et le graphe transposé de G . Utiliser une file.

Exercice 6. *Parcours « rapide ».* Écrire une fonction `compo g x` prenant en entrée un graphe g donné par listes d'adjacence, ainsi qu'un sommet x et renvoyant la liste des éléments dans la composante connexe de x . On pourra utiliser `List.mem` et on n'impose aucune complexité. Objectif : 5 lignes max.

2 Exercices plus théoriques

Exercice 7. *Suites graphiques.* Une suite finie d'entiers est dite graphique s'il existe un graphe non orienté dont les degrés des sommets correspondent à cette suite.

1. Les suites suivantes sont-elles graphiques ?

(3, 3, 2, 2) (3, 2, 1, 1) (3, 3, 1, 1) (3, 3, 2, 1, 1)

2. Trouver deux graphes non isomorphes correspondants à la suite (3, 2, 2, 2, 1) (deux graphes sont dits isomorphes s'il existe une bijection entre leurs sommets qui préserve les arêtes).

3. Soit $n \geq 2$ et $(d_1, d_2, \dots, d_n)$ une suite décroissante, avec $d_1 > 0$. Montrer l'équivalence des deux propriétés suivantes :

(i) la suite $(d_1, d_2, \dots, d_n)$ est graphique ;

(ii) la suite $(d_2 - 1, d_3 - 1, \dots, d_{d_1+1} - 1, d_{d_1+2}, d_{d_1+3}, \dots, d_n)$ est graphique.

4. En déduire un algorithme testant si une suite est graphique.

5. Exhiber un graphe correspondant à la suite (4, 4, 3, 2, 2, 1).

6. Dans un graphe associé à la suite (4, 4, 3, 2, 2, 1), les sommets de degré 2 peuvent-ils être voisins ?

Exercice 8. Un circuit dans un graphe orienté est dit *eulérien* lorsqu'il passe une et une seule fois par chaque arc du graphe. Un graphe orienté est dit eulérien s'il possède un circuit eulérien.

1. Prouver qu'un graphe orienté fortement connexe est eulérien si et seulement si chaque sommet a ses degrés entrant et sortant égaux.

2. Écrire en Caml une fonction permettant d'exhiber un circuit eulérien dans un graphe donné par listes d'adjacence vérifiant cette propriété. On essaiera si possible d'avoir une complexité en $O(|E|)$ (avec $|E|$ le nombre d'arcs). Le circuit sera donné sous la forme d'une liste.

Exercice 9. Codage de Prüfer d'un arbre. Considérons un arbre (au sens des graphes non orientés) $T = (V, E)$ avec $V = \{0, \dots, n-1\}$ et $n > 2$. Le codage de Prüfer permet de décrire précisément cet arbre à l'aide d'une suite finie de $n-2$ entiers de l'intervalle $\llbracket 0, n-1 \rrbracket$.

Algorithme 1 : Algorithme de Prüfer

Entrée : Un arbre $T = (V, E)$ (au sens des graphes non orientés)

Sortie : Une suite de $\llbracket 0, n-1 \rrbracket^{n-2}$

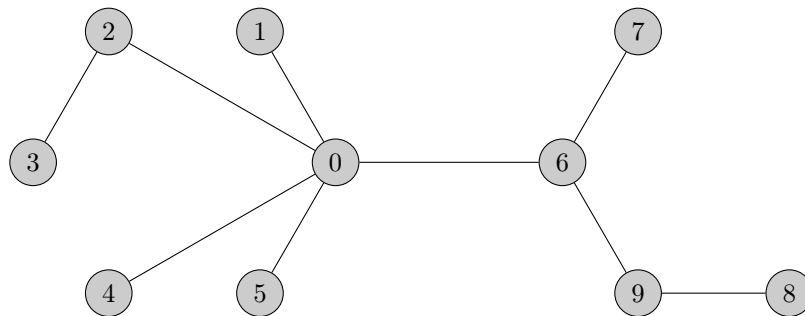
$L \leftarrow []$;

tant que $|V| > 2$ **faire**

$i \leftarrow \min\{k \in V \mid \deg(k) = 1\}$;
 $j \leftarrow$ l'unique k tel que $\{i, k\} \in E$;
 Rajouter j à L ;
 Supprimer i de V et $\{i, j\}$ de E

retourner L

1. Exécuter l'algorithme sur l'arbre suivant



2. Justifier que l'algorithme s'exécute sans erreur et retourne bien une liste de taille $n-2$.
3. Donner un arbre associé au codage de Prüfer $0, 0, 0, 0, 0, 0, 0$.
4. Montrer qu'un sommet s est de degré k dans l'arbre T si et seulement si s apparaît $k-1$ fois dans le codage de Prüfer de T .
5. Proposer une méthode pour reconstruire l'arbre T à partir de son codage de Prüfer (lu dans l'ordre dans lequel apparaissent les sommets) et son tableau des degrés.
6. Déterminer le nombre d'arbres différents sur l'ensemble de sommets $\llbracket 0, n-1 \rrbracket$.
7. Déterminer le nombre d'arbres différents sur l'ensemble de sommets $\llbracket 0, n-1 \rrbracket$, dont les degrés des sommets sont $d_0, \dots, d_{n-1}$.

Exercice 10. Diamètre d'un arbre. On appelle *diamètre* d'un graphe non orienté la longueur du plus long chemin acyclique que l'on peut effectuer dans le graphe. Montrer que l'algorithme suivant fournit le diamètre d'un arbre (au sens des graphes non orientés). Donner un contre-exemple sur un graphe connexe non acyclique.

Algorithme 2 : Diamètre d'un arbre

Entrée : Un arbre $G = (V, E)$ (au sens des graphes non orientés)

Sortie : Son diamètre

Prendre un sommet quelconque r , et déterminer un sommet u à distance maximale par un parcours en largeur ; Réeffectuer un parcours en largeur depuis u et renvoyer la distance maximale obtenue.

Exercice 11. Oral ENS. Soit G un graphe non orienté connexe. On note d_s le degré du sommet s . On place une pile de jetons sur chaque sommet du graphe ; le nombre de jetons sur le sommet s sera noté c_s .

On joue à un jeu dont la règle est la suivante : si $c_s \geq d_s$ pour un sommet s , on peut retirer d_s jetons au-dessus de s et en donner un à chaque sommet voisin de s . La partie se poursuit tant qu'il est possible de jouer l'un des sommets (on peut avoir des parties infinies...), si plusieurs sont jouables il n'y a pas de contrainte sur lequel jouer en premier.

1. Montrer que si une partie est infinie, tous les sommets ont été joués une infinité de fois.
2. Montrer que si une partie est finie, au moins un sommet n'a pas été joué.
3. Montrer que la condition $N \leq \sum (d_i - 1)$, avec N le nombre de jetons est nécessaire qu'une partie soit finie.
4. Cette condition est-elle suffisante ?
5. Montrer que pour une partie finie, toutes les autres parties jouées à partir de la même configuration de départ sont finies et parviennent à la même configuration finale. *Indication* : on considère une partie finie $C_0 \rightarrow C_1 \rightarrow \dots \rightarrow C_f$, où les C_i sont des configurations (nombre de jetons sur chaque sommet), et C_f la configuration finale (aucun sommet n'est jouable). Montrer qu'à partir de $C_0 \rightarrow \dots \rightarrow C_{i-1} \rightarrow C'_i \rightarrow \dots$ une autre partie où $C'_i \neq C_i$, on peut construire une partie $C_0 \rightarrow \dots \rightarrow C_{i-1} \rightarrow C_i \rightarrow C''_{i+1} \rightarrow \dots$, avec l'un des C''_j pour $j \geq i+1$ égal à l'un des C'_k .

Remarque : ce dernier résultat, non trivial, est appelé *confluence de la relation d'écoulement*.

3 Graphes planaires et colorations de graphe

Exercice 12. *Coloriage d'un graphe non orienté.* On dit qu'une fonction $f : V \rightarrow \mathbb{N}$ est un coloriage d'un graphe non orienté $G = (V, E)$ si pour tout $(i, j) \in E$, $f(i) \neq f(j)$. On dit que le graphe est k -colorable s'il existe un coloriage du graphe à valeurs dans $\llbracket 0, k-1 \rrbracket$. Le plus petit k qui convient est appelé le *nombre chromatique* du graphe.

1. Écrire une fonction `est_coloriage g f` prenant en entrée un graphe (V, E) donné par listes d'adjacence et f une fonction définie sur $\llbracket 0, n-1 \rrbracket$ (avec $n = |V|$) et retournant un booléen suivant si f est un coloriage du graphe ou non. Quelle est sa complexité si un appel à f s'effectue en temps constant ?
2. Quel est le nombre chromatique d'une clique à n sommets ?
3. Un sous-ensemble de sommets $S \subseteq V$ est appelé une clique du graphe si $\mathcal{P}_2(S) \subseteq E$. Comparer la taille maximale d'une clique du graphe avec son nombre chromatique.
4. Donner un exemple où l'inégalité est stricte.
5. Soit G un graphe tel que chaque sommet ait un degré au plus Δ . Montrer que le graphe admet une $(\Delta + 1)$ -coloration.
6. (*suite de la question précédente.*) Donner un exemple de graphe n'admettant pas de Δ -coloration.
7. Décrire sans coder un algorithme (naïf) testant si un graphe est k -colorable, pour $k \geq 2$, et estimer sa complexité en fonction de $|V|$, $|E|$ et k , si le graphe est encodé par listes d'adjacence.
8. Décrire un algorithme de complexité polynomiale testant si un graphe possède une 2-coloration (on s'inspirera du parcours en largeur). Donner sa complexité si le graphe est représenté par listes d'adjacence.

Tout comme le problème SAT, tester si un graphe admet une k -coloration est difficile, dès que $k \geq 3$, voir exercice 16. Calculer le nombre chromatique d'un graphe est donc un problème difficile.

Exercice 13. *Un algorithme glouton pour la coloration.* 1. Écrire une fonction `coloration_gloutonne g` prenant en entrée un graphe non orienté donné par listes d'adjacence, et renvoyant un coloriage du graphe. On appliquera une idée gloutonne : pour chaque sommet, lui donner la plus petite couleur non déjà donnée à un de ses voisins. On renverra un tableau `c` de taille n avec n l'ordre de `g`, tel que `c.(i)` contienne la couleur du i -ème sommet.

2. Quelle est la complexité ?
3. Montrer par un exemple que l'algorithme précédent ne fournit pas un coloriage optimal en nombre de couleurs.
4. Montrer toutefois que pour tout graphe, il existe un réordonnancement des sommets tel que l'application de l'algorithme glouton fournit un coloriage optimal.

Exercice 14. *Graphes planaires et formule d'Euler.* Un graphe non orienté est dit planaire s'il existe un plongement du graphe dans le plan, tel que deux arêtes ne se coupent jamais. Dans cet exercice, on autorise les boucles sur un sommet et les multi-arêtes. On peut supposer les graphes connexes lorsqu'on cherche des caractérisations des graphes planaires.

1. Soit $G = (V, E)$ un graphe plan, connexe. Le plongement du graphe dans le plan découpe celui-ci en zones appelées faces (ne pas oublier la face extérieure au graphe)¹. Démontrer la formule d'Euler : $|V| - |E| + |F| = 2$ (avec F l'ensemble des faces). On pourra raisonner par récurrence sur $|F|$.

1. Plus formellement, le complémentaire du graphe dans le plan est un ouvert pour la topologie usuelle. Les faces sont ses composantes connexes (par arcs).

- On définit le degré d'une face comme le nombre d'arêtes qui bordent cette face (pour une arête qui borde deux fois une face, cette arête compte double dans le degré). Montrer que $\sum_{f \in F} \deg(f) = 2|E|$.
- Cette formule implique que le degré moyen d'une face est $2|E|/|F|$. En déduire que le graphe complet K_5 et le graphe biparti complet $K_{3,3}$ ne sont pas planaires.

Le mathématicien polonais Kazimierz Kuratowski a établi en 1930 qu'un graphe qui ne contient pas de sous graphe qui est une expansion de K_5 ou $K_{3,3}$ est planaire. Une expansion d'un graphe G s'obtient en rajoutant des sommets de degré 2 le long des arêtes de G .

Exercice 15. Graphes planaires et 5-coloration. Le but de cet exercice est de montrer que les graphes planaires sont 5-colorables. On peut toujours se restreindre aux graphes connexes dans un problème de coloration, on considère donc dans tout l'exercice des graphes planaires connexes. On remarque de plus que supprimer une boucle ou une multi-arête ne change rien aux questions de colorabilité. On suppose donc que les graphes considérés sont *simples* : ils n'ont ni boucles, ni multi-arêtes. On reprend les notations de l'exercice 14, et on rappelle que pour un graphe planaire (simple) $G = (V, E)$, on a $|F| - |E| + |V| = 2$, et $\sum_{f \in F} \deg(f) = 2|E|$.

- Montrer que si un graphe planaire simple $G = (V, E)$ possède au moins 3 arêtes, alors $2|E| \geq 3|F|$.
- Montrer que le degré moyen d'un sommet dans un graphe planaire simple est strictement inférieur à 6.
- En déduire qu'un graphe planaire est 6-colorable.
- On va montrer par récurrence sur $|V|$ pour que le graphe est 5-colorable. L'initialisation est évidente pour $|V| \leq 5$. On se donne donc un graphe planaire connexe ayant au moins 6 sommets. La question 2 montre qu'il existe dans le graphe un sommet de degré au plus 5. Conclure si le graphe possède un sommet de degré au plus 4.
- C'est l'étape difficile de la démonstration : si le graphe ne possède pas de sommet de degré au plus 4, il possède néanmoins un sommet s de degré 5, dont les voisins sont notés $x_1, \dots, x_5$.
 - Justifier qu'au moins deux des voisins de s ne sont pas voisins. On suppose sans perte de généralité que ce sont x_1 et x_2 (indication : voir exercice précédent !)
 - On considère le graphe G' obtenu en *contractant* les sommets s, x_1 et x_2 en un unique sommet s' : toutes les arêtes de la forme $\{v, s\}$, $\{v, x_1\}$ ou $\{v, x_2\}$ avec $v \notin \{s, x_1, x_2\}$ dans G mènent à une unique arête $\{v, s'\}$ dans G' (on admet que G' est planaire). Par hypothèse de récurrence, G' est 5-colorable. Expliquer comment obtenir une 5-coloration c de G à partir d'une 5-coloration c' de G' .

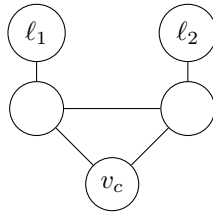
Les graphes planaires sont en fait 4-colorables : c'est le fameux « théorème des 4 couleurs ». Mais il n'existe pas de démonstration « simple » de ce résultat : la preuve repose sur l'examen (informatique) de 1478 cas critiques.

4 Problème 3-SAT et 3-coloration

Exercice 16. 3-COL est « aussi difficile » que 3-SAT. On a dit dans le cours de logique que le problème de déterminer si une expression logique sous forme de conjonction de clauses à 3 littéraux est satisfiable (problème 3-SAT) est un problème difficile². Le but de cet exercice est de montrer que le problème de décider si un graphe non orienté admet une 3-coloration (problème 3-COL, voir exercice 12 pour les définitions) est également difficile. La technique utilisée est classique : elle consiste à *réduire* un problème déjà connu comme difficile (ici 3-SAT) au problème dont on veut montrer la difficulté (ici 3-COL). Pour ce faire, il faut montrer que toute instance de 3-SAT (une formule logique conjonction de clauses à trois littéraux) peut être transformée *en temps polynomial* en une instance de 3-COL (un graphe non orienté), et que la formule logique est satisfiable si et seulement si le graphe est 3-colorable : cette transformation montre que s'il existe un algorithme polynomial résolvant 3-COL, il en existe un résolvant 3-SAT. La démarche va être la suivante : on va transformer une instance de 3-SAT (conjonction de clauses à trois littéraux qu'on supposera à variables distinctes) en un graphe ayant notamment pour sommets les variables logiques intervenant dans la formule et leurs négations. À chaque clause sera également associée un sommet, et traduire chaque clause se fera via l'introduction de nouveaux sommets. On utilisera également deux sommets reliés A et B . Dans l'idée de la construction qui suit, la couleur 1 est associée à vrai et la couleur 0 à faux.

- On considère le graphe :

2. Rappel : il n'existe à l'heure actuelle pas d'algorithme de complexité (dans le pire cas) polynomiale en la taille d'une expression logique permettant de décider si elle est satisfiable. Il n'en existera jamais si $P \neq NP$.



Montrer que le graphe a les deux propriétés suivantes.

- Dans tout 3-coloriage du graphe avec les couleurs $\{0, 1, 2\}$, si ℓ_1 et ℓ_2 sont de couleurs autre que 2, et si v_c est de couleur 1, alors ℓ_1 ou ℓ_2 est de couleur 1.
- Réciproquement, pour un choix de coloriage de ℓ_1 et ℓ_2 avec les couleurs 0 ou 1, si l'un au moins des ℓ_i est colorié avec 1, il existe un prolongement de la coloration au graphe entier avec v_c de couleur 1.

Intuitivement, ce graphe encode une disjonction de deux littéraux.

2. À partir de l'exemple précédent, donner un graphe ayant notamment 4 sommets ℓ_1, ℓ_2, ℓ_3 et v_c , vérifiant les deux propriétés suivantes.

- dans tout 3-coloriage du graphe avec les couleurs $\{0, 1, 2\}$, si ℓ_1, ℓ_2 et ℓ_3 sont de couleurs autre que 2, et si v_c est de couleur 1, alors l'un au moins des ℓ_i est de couleur 1.
- Réciproquement, pour un choix de coloriage de ℓ_1, ℓ_2 et ℓ_3 avec les couleurs 0 ou 1, si l'un au moins des ℓ_i est colorié avec 1, il existe un prolongement de la coloration au graphe entier avec v_c de couleur 1.

Indication : $\ell_1 \vee \ell_2 \vee \ell_3 = (\ell_1 \vee \ell_2) \vee \ell_3$.

3. Trouver un graphe ayant au moins deux sommets v et $\neg v$, tel que dans tout 3-coloriage, si v et $\neg v$ sont de couleurs différentes de 2, alors v est de couleur 1 si et seulement si $\neg v$ n'est pas de couleur 1 (ceci encode la négation).
4. Si un graphe possède deux sommets adjacents A et B de couleur 0 et 2, dans un 3-coloriage du graphe, quelles sont les couleurs des sommets adjacents à la fois à A et B ? (Ceci encode une conjonction).
5. Dédurre des trois questions précédentes qu'il existe un algorithme en temps polynomial en la taille de l'entrée, qui transforme tout ensemble S de 3-clauses en un graphe G qui est 3-colorable si et seulement si S est satisfiable.
Indication : on créera deux sommets distingués A et B dans G , adjacents; par symétrie, on supposera que A est de couleur 0 et B de couleur 2. De plus, avec n le nombre de variables logiques, on introduira $2n$ sommets (pour chaque v et $\neg v$). Pour chaque clause c , on introduira un sommet v_c et des sommets permettant de relier les sommets associés aux littéraux apparaissant dans la clause c à v_c à la manière du graphe de la question 2. Indiquer les arêtes à rajouter.

Comme 3-SAT est NP-complet, la question 5 montre que le problème 3-COL est NP-difficile : au moins aussi difficile que tout problème de la classe NP. À partir d'un graphe $G = (V, E)$ donné et d'une fonction $f : V \rightarrow \{0, 1, 2\}$, il est facile (i.e faisable en temps polynomial, voir exercice 12) de vérifier si f est un 3-coloriage du graphe, ce qui montre que 3-COL est dans NP. On a donc montré que 3-COL est lui aussi NP-complet.