Université Clermont Auvergne
BUT Info 1A
R207 - Graphes
Année 2021-2022

Florent Foucaud
Malika More
Thibault Ralet
Carine Simon
Thierry Trévisan

TD7 - Graphes d'intervalles – ordres topologiques

Les exercices commencent à la page 4

Rappel de cours

Coloration

Définition. Une *coloration propre* d'un graphe G non-orienté est une fonction qui attribue à chaque sommet de G une valeur (appelée couleur) de telle façon que deux sommets adjacents (reliés par une arête) reçoivent deux couleurs différentes.

Le *nombre chromatique* d'un graphe G, noté $\chi(G)$, est le plus petit nombre de couleurs possible dans une coloration de G.

Graphe orienté acyclique

Définition. Un **graphe orienté acyclique** (en anglais, **DAG** : *directed acyclic graph*) est un graphe orienté qui ne possède aucun cycle (c'est-à-dire, une chaîne orientée partant d'un sommet pour y revenir).

Le graphe de gauche est acyclique, tandis que celui de droite possède un cycle (6 - 7 - 8).

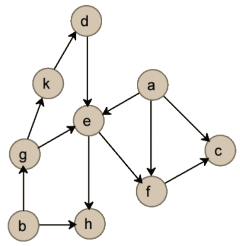
Typiquement, un graphe orienté acyclique représente un *problème d'ordonnancement de tâches* : un sommet correspond à une tâche et un arc (u, v) indique la contrainte de précédence « *la tâche u doit être terminée avant que la tâche v ne puisse commencer* ». Nous verrons des exemples dans les exercices.

Ordre topologique

Définition. Soit G un graphe orienté acyclique. Un ordre $<$ sur les sommets est dit **topologique** s'il *respecte toutes les arêtes du graphe.* C'est-à-dire :

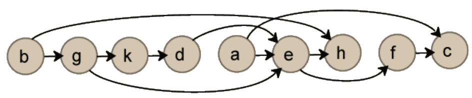
$$\text{Pour tout arc } (u, v) \text{ de } G, \text{ on a } \quad u < v.$$

Exemple : Voici un graphe orienté acyclique, et un ordre topologique correspondant.



$$b < g < k < d < a < e < h < f < c$$

Graphiquement, il est très facile de voir qu'il s'agit bien d'un ordre topologique, car les arcs sont tous orientés de la gauche vers la droite.



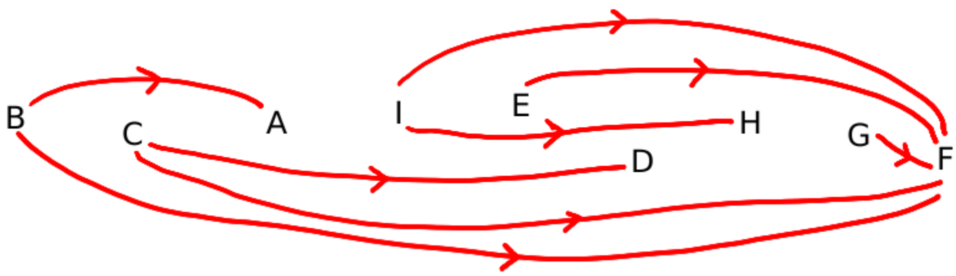
Remarque 1 : Un même graphe peut admettre *plusieurs ordres topologiques* différents.

Remarque 2 : Évidemment, si le graphe contient un cycle, il est impossible de trouver un ordre topologique sur les sommets.

Algorithme pour les humains et les petits graphes

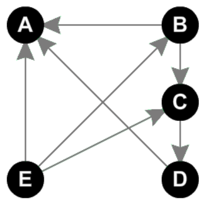
1. Trouver dans le graphe un sommet S qui n'a pas de prédécesseur
2. Placer S à la suite de la liste ordonnée des sommets obtenue jusqu'ici
3. Supprimer le sommet S du graphe
4. Recommencer tant qu'il y a des sommets dans le graphe
5. La liste ordonnée des sommets ainsi obtenue est un ordre topologique pour le graphe initial

Par exemple : B - C - A - I - E - D - H - G - F

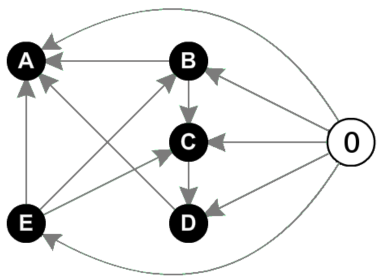


Algorithme plus efficace basé sur un parcours en profondeur

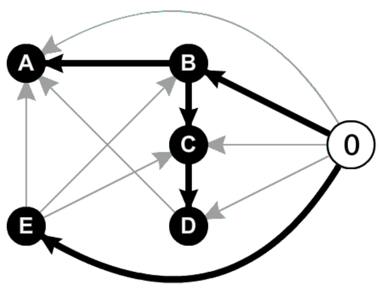
Étant donné un graphe orienté acyclique G, on peut toujours lui trouver un ordre topologique, avec la méthode suivante.



Tout d'abord, on rajoute un sommet « source » connecté à tous les sommets du graphe. Par exemple, pour le graphe précédent, on rajoute le sommet « 0 » (à droite) :



Puis on effectue un **parcours en profondeur** sur le nouveau graphe, au départ du sommet source. Un résultat possible est représenté ci-dessous. Lors du parcours en profondeur, on liste les sommets **dans l'ordre où l'algorithme finit de les traiter**. Dans cet exemple : A, D, C, B, E.

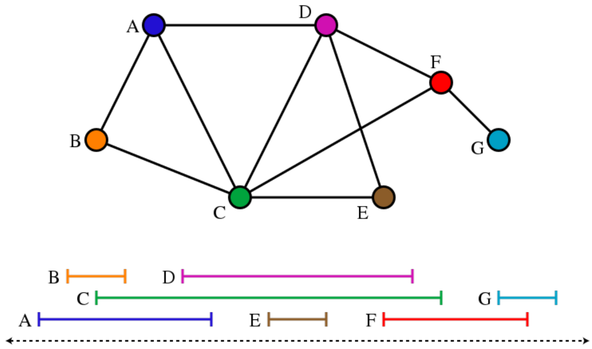


Alors, en prenant les sommets dans l'ordre inverse, on obtient un ordre topologique. Soit ici :

$$E < B < C < D < A.$$

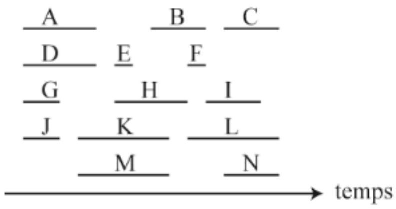
Exercice 1 (Graphes d'intervalles)**.**

Dans cet exercice, on s'intéresse à la coloration des **graphes d'intervalles**. Un graphe d'intervalles est le graphe d'intersection d'un ensemble d'intervalles de la droite réelle. Chaque sommet du graphe d'intervalles représente un intervalle de l'ensemble, et une arête relie deux sommets lorsque les deux intervalles correspondants s'intersectent. Voici un exemple :

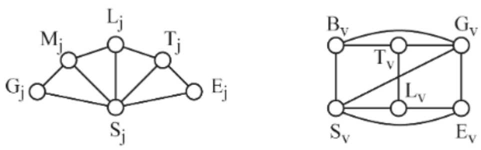


Les graphes d'intervalles sont utilisés pour modéliser les problèmes d'allocation de ressources. Chaque intervalle représente une demande pour une ressource (telle qu'une unité de traitement d'un système informatique distribué ou une salle de classe) pour une période de temps spécifique. Une coloration optimale des sommets du graphe d'intervalles représente une affectation de ressources qui couvre toutes les demandes avec le moins de ressources possible (1 couleur = 1 ressource). La coloration du graphe ci-dessus n'est pas optimale.

1. **Construction d'un graphe d'intervalles.** Dessiner le graphe d'intervalles correspondant aux tâches ci-dessous :



2. **Reconnaissance d'un graphe d'intervalles.** Déterminer si un graphe donné est ou pas un graphe d'intervalles n'est pas une question évidente. Ainsi, parmi les deux graphes ci-dessous, l'un est un graphe d'intervalles et pas l'autre.



   (a) Montrer qu'une chaîne et une clique sont des graphes d'intervalles et qu'un cycle à 4 sommets (ou plus) n'est pas un graphe d'intervalles.

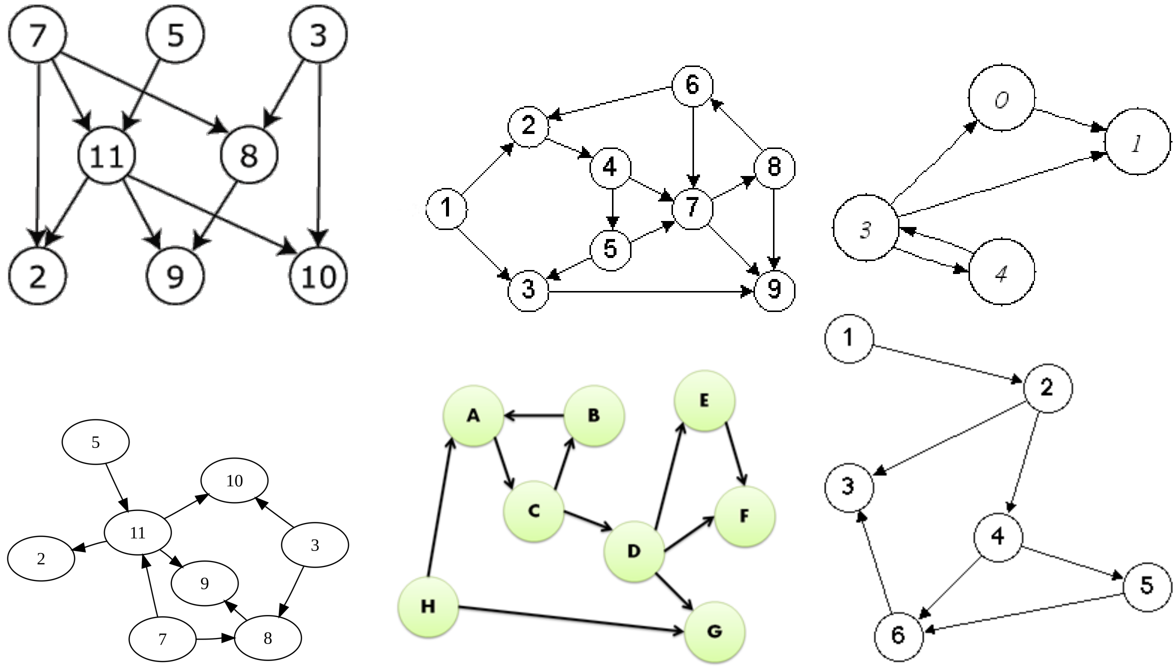
   (b) Déterminer, en justifiant votre réponse, lequel des deux graphes ci-dessus **n'est pas** un graphe d'intervalles. Donnez des intervalles correspondant à l'autre graphe.

3. **Coloration gloutonne d'un graphe d'intervalles** On utilise l'algorithme de coloration gloutonne suivant : on trie les intervalles par dates de début, puis pour chaque intervalle, on lui assigne la plus petite couleur non-utilisée par un intervalle commençant précédemment et qui le chevauche. On s'assure ainsi que deux sommets représentant des intervalles voisins n'ont pas la même couleur. Il s'agit donc bien d'un algorithme de coloration.

   (a) Coloriez le graphe obtenu à la première question en utilisant cet algorithme.

   (b) Montrez que le coloriage obtenu par cette méthode gloutonne sur un graphe d'intervalles est toujours optimal, c'est-à-dire qu'il utilise le plus petit nombre de couleurs possible.

Exercice 2. Parmi les graphes ci-dessous, lesquels sont des graphes orientés acycliques ?

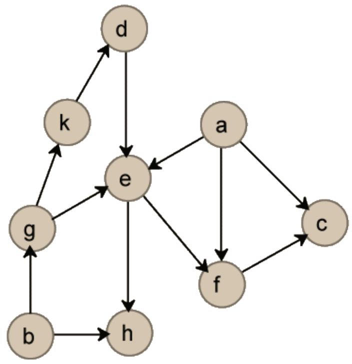


Exercice 3. Voici trois ordres possibles pour les sommets. Le(s)quel(s) sont des ordres topologiques pour le graphe ?

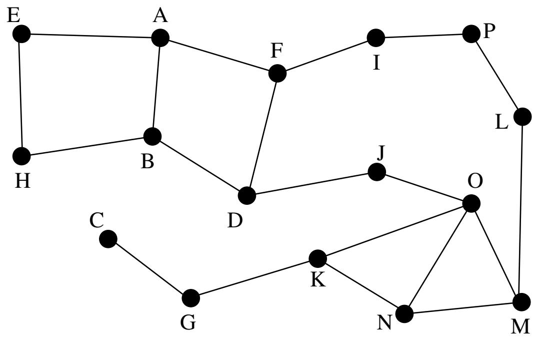
a) $a < b < c < d < e < f < g < h < k$

b) $b < g < d < a < k < e < h < c < f$

c) $a < b < g < k < d < e < h < f < c$



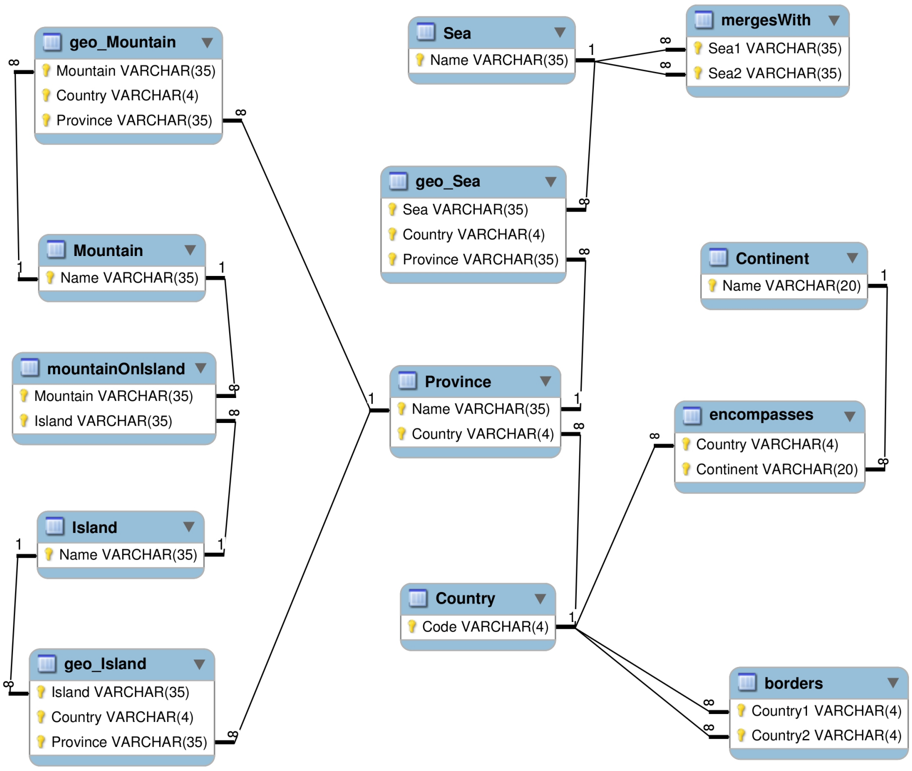
Exercice 4. Soit le graphe suivant :



1. Orientez les arcs du graphe selon l'ordre alphabétique. Par exemple, pour l'arête qui se trouve entre les sommets E et A, vous mettrez une flèche de A *vers* E, parce-que A est avant E dans l'ordre alphabétique. Justifiez que le graphe orienté obtenu G est acyclique.
2. Avec l'algorithme de suppression des sommets sans prédécesseur, trouvez un ordre topologique pour G.
3. Même question avec l'algorithme basé sur un parcours en profondeur.

Exercice 5. On considère la base de données dont le schéma relationnel est donné en notation UML ci-dessous [1]. Une référence va de la fille vers la mère. On ne peut pas supprimer les données d'une table auxquelles une autre table fait référence. Pour garantir la suppression de données sans incident, on procède toujours à la suppression de données en commençant par la fille puis par la mère.

1. Notation UML : la table fille est du côté marqué ∞, la table mère est du côté marqué 1.



1. Construire un graphe orienté sans cycle dont un ordre topologique représente un ordonnancement valide des destructions des tables de la base de données (utiliser les noms des tables pour nommer les sommets).
2. En utilisant l'algorithme basé sur un parcours en profondeur, calculer un ordre topologique et en déduire un ordre de suppression des données correct. Lorsque un choix se présentera pendant le parcours, on choisira le sommet le plus petit dans l'ordre alphabétique. Rédigez votre réponse en indiquant pour chaque sommet le temps de première visite, des visites intermédiaires (si c'est le cas) et de dernière visite.

Exercice 6. Un dandy doit s'habiller. Pour être fin prêt, il doit mettre :

- son caleçon,
- ses chaussettes,
- ses chaussures,
- sa chemise,
- ses boutons de manchettes,
- sa ceinture,
- sa cravate,
- son épingle à cravate,
- son pantalon,
- son gilet,
- sa montre à gousset,
- sa veste,
- sa pochette (son vilain petit mouchoir),
- son chapeau.

Naturellement, il ne peut pas s'habiller dans n'importe quel ordre. En effet :

- il doit mettre son caleçon avant son pantalon et ses chaussures ;
- ses chaussures ne peuvent être enfilées avant les chaussettes ;

– la chemise doit être mise avant la ceinture, les boutons de manchettes, le chapeau et la cravate ;
– la cravate doit être nouée avant de pouvoir être fixée par l'épingle à cravate ;
– la cravate doit être nouée avant de fermer le gilet ;
– la montre-à-gousset se porte sur le gilet ;
– la veste ne peut être mise avant la montre-à-gousset ;
– la ceinture ou les chaussures ne peuvent pas être mises avant le pantalon.

Modélisez ce délicat problème et résolvez-le à l'aide de l'algorithme basé sur un parcours en profondeur.

Exercice 7. Au cours de la rénovation de son habitation, la famille M. a décidé de transformer en cuisine une ancienne remise. Le tableau ci-dessous énumère les différentes tâches et les contraintes d'antériorité entre ces tâches.

Désignation	Libellé	Tâches préalables
A	Plomberie	-
B	Gaines électriques	-
C	Isolant plafond	F
D	Isolant murs	E
E	Rails de fixation des plaques de plâtre des murs	A,B
F	Rails de fixation des plaques de plâtre du plafond	A,B
G	Plaques de plâtre des murs	D,M
H	Plaques de plâtre du plafond	C
I	Carrelage	G
J	Peinture	G,H
K	Meubles et plans de travail	I,J,L
L	Prises électriques et interrupteurs	G
M	Fenêtres	-

1. Construire le graphe des tâches associé.
2. Effectuer un parcours en profondeur sur le graphe des tâches. En cas d'ambiguité, on choisira les sommets dans l'ordre alphabétique. Pour chaque sommet, on précisera l'instant de découverte et l'instant de clôture.
3. En déduire un ordre topologique de réalisation des tâches qui respecte les contraintes d'antériorité.

Exercice 8. Prouvez qu'un graphe orienté avec cycle n'admet jamais d'ordre topologique.

Exercice 9 (**)**.** Le but de cet exercice est de caractériser les graphes orientés acycliques ne possédant qu'un seul ordre topologique.

1. On commence par un sous-cas relativement simple : Caractérisez les *arbres* orientés ne possédant qu'un seul ordre topologique.
2. Caractérisez les graphes orientés acycliques ne possédant qu'un seul ordre topologique.