# Applications des graphes à l'ordonnancement

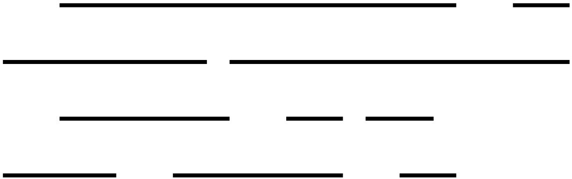
Florent Foucaud - Malika More - Thibault Ralet
Carine Simon - Thierry Trévisan

BUT Info 1A - UCA

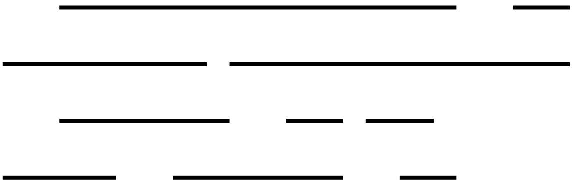
R207 Graphes

Année 2021-2022

# Contraintes temporelles



# Contraintes temporelles

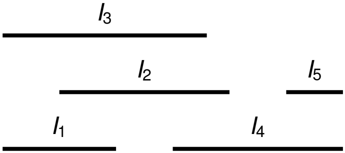


- tâches sur un supercalculateur
- cultures agricoles
- cours à l'IUT
- données archéologiques

# Graphe d’intervalles

## Définition

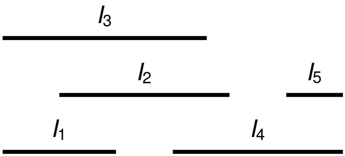
À chaque intervalle correspond un sommet du graphe. Deux sommets sont reliés par une arête si leurs intervalles s’intersectent.

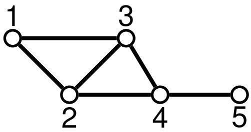


# Graphe d'intervalles

## Définition

À chaque intervalle correspond un sommet du graphe. Deux sommets sont reliés par une arête si leurs intervalles s'intersectent.

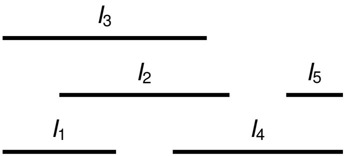


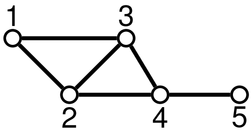



# Graphe d'intervalles

## Définition

À chaque intervalle correspond un sommet du graphe. Deux sommets sont reliés par une arête si leurs intervalles s'intersectent.

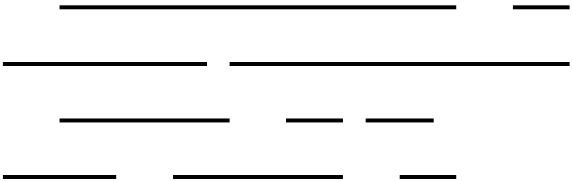



Coloration propre du graphe : affectation de ressources aux objets temporels

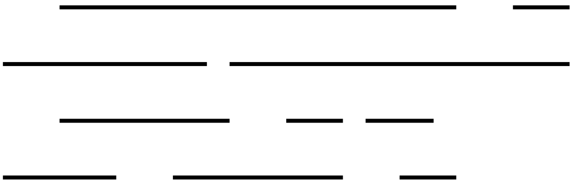
# Comment colorer le graphe d'intervalles ?

On utilise l'algorithme glouton de gauche à droite.



# Comment colorer le graphe d'intervalles ?

On utilise l'algorithme glouton de gauche à droite.



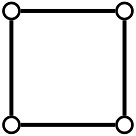
**Théorème**

*Dans un graphe d'intervalles G, on a toujours $\chi(G) = \omega(G)$ et l'algorithme glouton est optimal.*

# Quels graphes sont des graphes d'intervalles ?

**Question.**

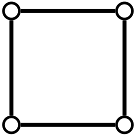
Ce graphe est-il un graphe d'intervalles ?



# Quels graphes sont des graphes d'intervalles ?

**Question.**

Ce graphe est-il un graphe d'intervalles ?



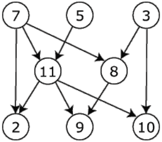
**Théorème (Lekkerkerker-Boland 1962 - Hors programme)**

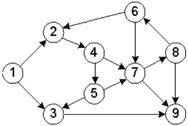
*Un graphe est d'intervalles si et seulement si il ne contient pas de cycle induit de longueur au moins 4 ni de triplet astéroïde.*

# Graphe orienté acyclique

## Définition

Un **graphe orienté acyclique** (**DAG** : *directed acyclic graph*) est un graphe orienté qui ne possède aucun cycle (une chaîne orientée partant d'un sommet pour y revenir).



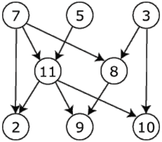



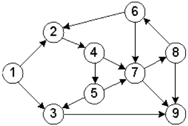

Le graphe de gauche est acyclique, tandis que celui de droite possède un cycle (6 - 7 - 8).

# Graphe orienté acyclique

## Définition

Un **graphe orienté acyclique** (**DAG** : *directed acyclic graph*) est un graphe orienté qui ne possède aucun cycle (une chaîne orientée partant d'un sommet pour y revenir).



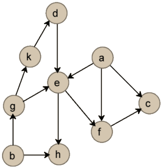

*problème d'ordonnancement de tâches* : un sommet correspond à une tâche et un arc $(u, v)$ indique la contrainte de précédence « *la tâche u doit être terminée avant que la tâche v ne puisse commencer* »

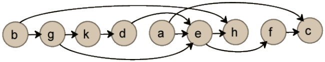
# Ordre topologique

## 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 :

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



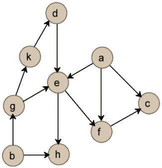

$$b < g < k < d < a < e < h < f < c$$

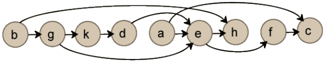
# Ordre topologique

## 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 :

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



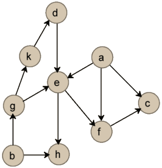

**Les arcs sont tous orientés de la gauche vers la droite**

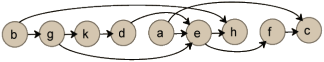
# Ordre topologique

## 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 :

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



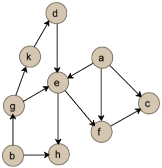

Un même graphe peut admettre *plusieurs ordres topologiques* différents

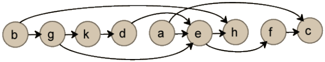
# Ordre topologique

## 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 :

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

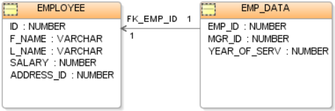



Si le graphe contient un cycle, il est **impossible** de trouver un ordre topologique sur les sommets

# Clés étrangères en bases de données

- Dans les bases de données relationnelles, il existe des contraintes entres les tables
- Des colonnes d'une table font référence à des colonnes d'une autre table
- Ces contraintes permettent de garantir l'intégrité des données
- On parle de *clés étrangères*.



# Suppression des tables dans une base de données

**Problème.**

On désire supprimer des tables dans une base de données

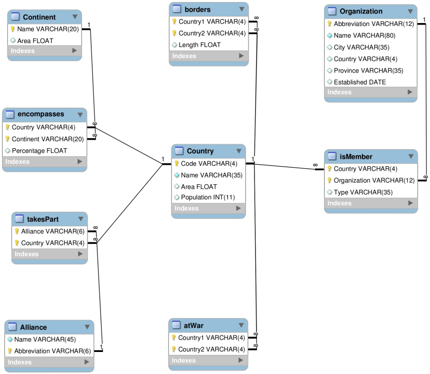
**Obstacle.**

Lorsqu'une table est référencée dans une autre table à l'aide d'une clé étrangère, on ne peut pas la détruire

**Question.**

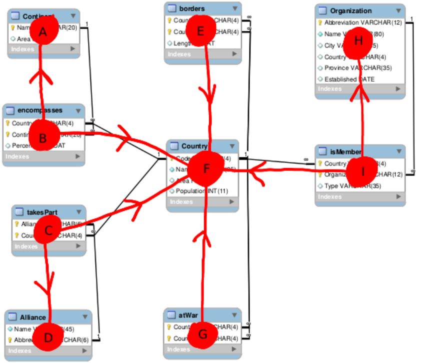
Peut-on toujours trouver un ordre adapté pour supprimer des tables dans une base ayant des clés étrangères ? Si c'est le cas, comment faire pour trouver un tel ordre ?

# Exemple




- Notation UML : la table fille est du côté marqué $\infty$, la table mère du côté marqué 1.
- La référence va de la fille vers la mère (de $\infty$ vers 1).

# Bases données et ordre topologique

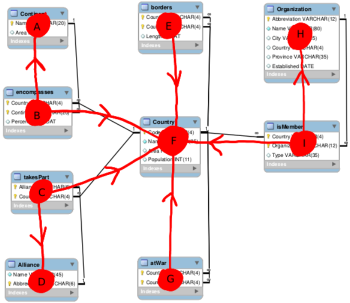


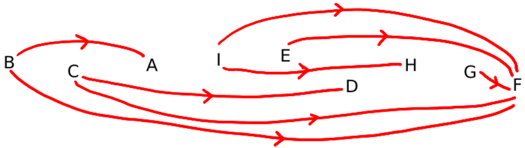

Un graphe orienté sans cycle dont un ordre topologique représente un ordonnancement valide des destructions des tables de la base de données

# ... 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

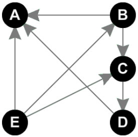
# Exemple (voir vidéo du calcul)

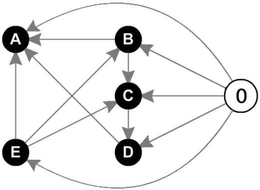


# Avant de commencer

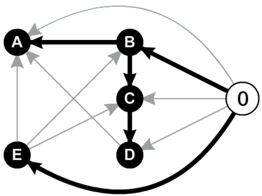
On ajoute un sommet « source » connecté à tous les sommets du graphe



# Algorithme plus efficace

On effectue un **parcours en profondeur** au départ du sommet source




En prenant les sommets **dans l'ordre inverse où l'algorithme finit de les traiter**, on obtient un ordre topologique :

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

**Algorithme 1 :** Parcours en profondeur + Tri Topologique

**Variables locales**

*S un ensemble, P une pile*

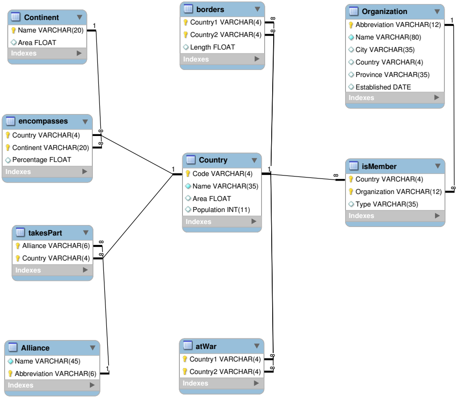
*Priorité un tableau* /* indexé par les sommets */

**début**

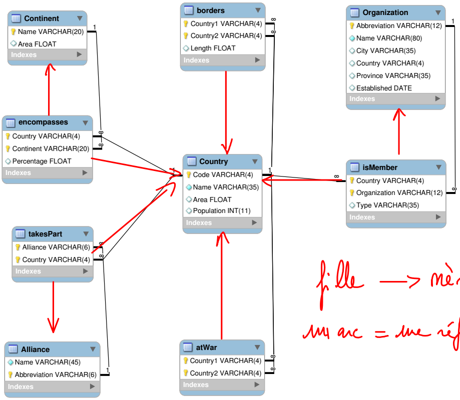
> **initialisation**
>
> *x=0*
>
> `ajouter(`$s$,$S$`)`
>
> `empiler(`$s$,$P$`)`
>
> **répéter**
>
> > $x := x + 1$
> >
> > $v$ :=`valeurTetePile`($P$)
> >
> > **si** *il existe* $u \notin S$ *successeur de* $v$ **alors**
> >
> > > `ajouter(`$u$,$S$`)` /* première visite de $u$ */
> > >
> > > `empiler(`$u$,$P$`)`
> >
> > **sinon**
> >
> > > `dépiler(`$P$`)` /* dernière visite de $v$ */
> > >
> > > *Priorité*$[v] := x$
> >
> > **fin si**
>
> **jusqu'à** `estVide(`$P$`)`

**fin**

# Exemple


Continent
Name VARCHAR(20)
Area FLOAT
Indexes
borders
Country1 VARCHAR(4)
Country2 VARCHAR(4)
Length FLOAT
Indexes
Organization
Abbreviation VARCHAR(12)
Name VARCHAR(80)
City VARCHAR(35)
Country VARCHAR(4)
Province VARCHAR(35)
Established DATE
Indexes
encompasses
Country VARCHAR(4)
Continent VARCHAR(20)
Percentage FLOAT
Indexes
Country
Code VARCHAR(4)
Name VARCHAR(35)
Area FLOAT
Population INT(11)
Indexes
isMember
Country VARCHAR(4)
Organization VARCHAR(12)
Type VARCHAR(35)
Indexes
takesPart
Alliance VARCHAR(6)
Country VARCHAR(4)
Indexes
Alliance
Name VARCHAR(45)
Abbreviation VARCHAR(6)
Indexes
atWar
Country1 VARCHAR(4)
Country2 VARCHAR(4)
Indexes

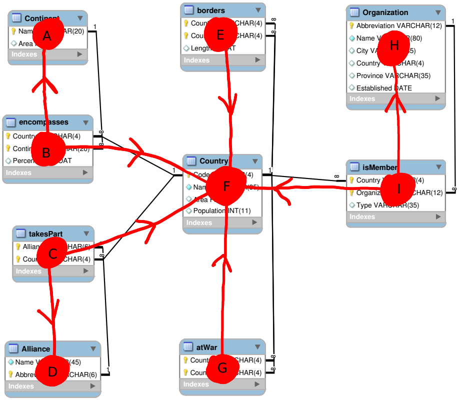
# Exemple



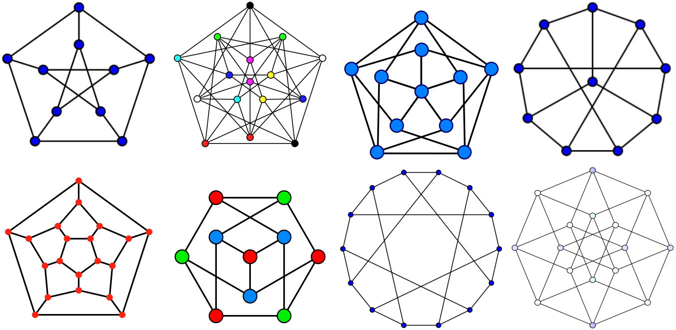

fille ⟶ mère

un arc = une référence.

# Exemple


Continent
Indexes
encompasses
Indexes
takesPart
Indexes
Alliance
Indexes
borders
Indexes
Country
Population INT(11)
Indexes
atWar
Indexes
Organization
Abbreviation VARCHAR(12)
Province VARCHAR(35)
Established DATE
Indexes
isMember
Indexes
A
B
C
D
E
F
G
H
I

# N'oubliez pas : les graphes sont partout !



Merci et au revoir !