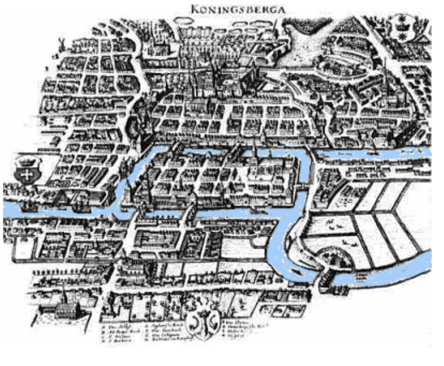
2 Graphes : notions de base

- Les 7 ponts de Königsberg
- Que sont les graphes
- A quoi servent les graphes
- Des définitions…
- Des exercices…
- Représentation des graphes en machine

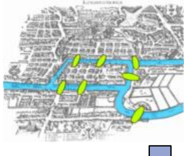
Les 7 ponts de Königsberg



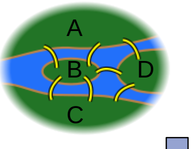
Étant donné que la ville est construite sur deux îles reliées au continent par six ponts, et entre elles par un pont, trouver un chemin quelconque permettant, à partir d'un point de départ au choix, de passer une et une seule fois par chaque pont, et de revenir à son point de départ, étant entendu qu'on ne peut traverser l'eau qu'en passant par les ponts.

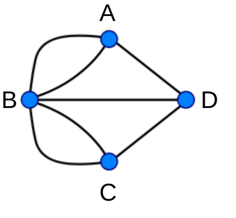
7 ponts de Königsberg

- Modélisation du problème à l'aide d'un graphe



Leonhard Euler
1707 - 1783





7 ponts de Königsberg

- 7 ponts de Königsberg
  - recherche d'un cycle eulérien (chaîne passant par toutes les arêtes du graphe une et une seule fois, et revenant à son point de départ)
    - possible si et seulement si le graphe associé au problème ne possède aucun sommet de degré impair
  - le graphe représentant Königsberg possède 4 sommets de degrés impairs (chaque berge est le départ de 3 ou 5 ponts)
    - le problème n'admet donc pas de solution
- Simplification du problème
  - emprunter chacun des ponts une seule fois sans pour autant revenir à son point de départ
    - recherche d'une chaîne eulérienne (chaîne passant par toutes les arêtes du graphe une et une seule fois)
    - possible que si le graphe associé au problème possède 0 ou 2 sommets de degré impair
    - dans le deuxième cas, le chemin doit partir d'un des sommets de degré impair, et aboutir à l'autre sommet de degré impair

Que sont les graphes ?

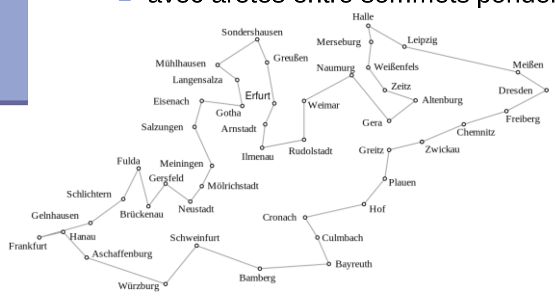
- Des objets mathématiques
  - un ensemble de nœuds ou sommets
  - mis en relation par des arcs ou des arêtes
  - orienté ou non
- Disposant d'une représentation « graphique »
  - d'où leur nom
  - algorithmes originaux guidés par l'aspect graphique

A quoi servent les graphes

- Modélisation de réseaux
  - transport
  - fluides
  - énergie
  - informatique, Web…
  - réseaux sociaux
- Représentation de liens logiques
  - dépendances entre fichiers sources
  - héritage
- Modélisation de l'état d'un système
- Ordonnancement de tâches

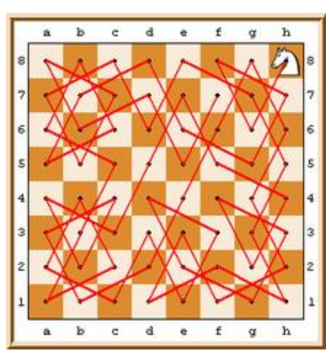
Quelques problèmes « célèbres »

- Problème du voyageur de commerce [travel(l)ing salesman problem]
  - un voyageur de commerce doit passer par toutes les villes
  - en minimisant la distance parcourue
- Recherche de cycle hamiltonien
  - avec arêtes entre sommets pondérées



Quelques problèmes « célèbres »

- Déplacement d'un cavalier sur un échiquier
  - un cavalier peut-il passer une et une seule fois sur chaque case et revenir à son point de départ ?
  - variantes
    - forme de l'échiquier
    - taille…



Définition d'un graphe non-orienté

- Graphe non-orienté **G = (X,U)**
- Sommet(s) [vertex (vertices)]
  - **X** ensemble des sommets du graphe **G**
    - $X = \{x_1, x_2, \ldots, x_n\}$
    - $|X| = n$ fini
- Arêtes [edges]
  - **U** ensemble des arêtes du graphe **G**
    - $u_i \in U$ est un couple **(origine, extrémité)**
    - **origine** $\in$ **X**, **extrémité** $\in$ **X**

Définition d'un graphe orienté

- Graphe orienté **G = (X,U)**
- Sommets
  - **X** ensemble des sommets du graphe **G**
    - $X = \{x_1, x_2, \ldots, x_n\}$
    - $|X| = n$ fini
- Arcs [directed edges]
  - **U** ensemble des arcs du graphe **G**
    - $u_i \in U$ est un couple **(origine, extrémité)**
    - **origine** $\in$ **X**, **extrémité** $\in$ **X**

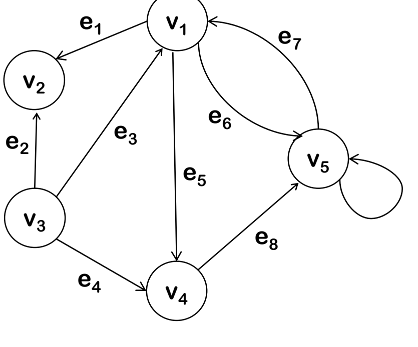
Représentation d'un graphe

- Ceci est un graphe !
  - **X = {v_1, v_2, v_3, v_4, v_5}**
  - **U = {(v_1, v_2), (v_1, v_4), (v_1, v_5), (v_3, v_1), (v_3, v_2), (v_3, v_4), (v_4, v_5), (v_5, v_1), (v_5, v_5)}**

- La complexité des algorithmes sera généralement fonction de **|X|** et/ou **|U|**

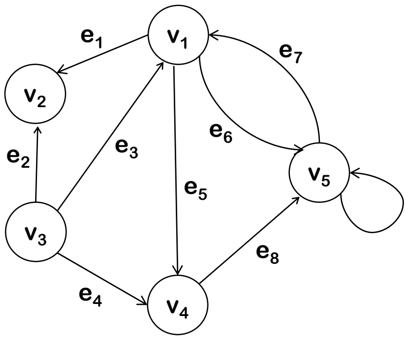
Représentation d'un graphe

- Ceci est un graphe !
  - **X = {v_1, v_2, v_3, v_4, v_5}**
  - **U = {(v_1, v_2), (v_1, v_4), (v_1, v_5), (v_3, v_1), (v_3, v_2), (v_3, v_4), (v_4, v_5), (v_5, v_1), (v_5, v_5),}**



Représentation d'un graphe

- La représentation graphique aide à comprendre nombre de leurs propriétés
- Que peut-on « voir » sur la représentation graphique ?
  - connexité
  - symétrie
  - « clusters »
  - distances…



Graphes simples, multigraphes

- Un graphe simple [simple graph]
  - une seule arête (ou un seul arc) entre deux sommets donnés
  - pas de boucles
- Un multigraphe [multigraph]
  - plusieurs arêtes peuvent exister entre deux mêmes sommets

Adjacence, degré

- L'adjacence
  - 2 sommets sont dits adjacents [adjacent] s'ils sont reliés par un arc
- Le degré [degree] ou arité d'un sommet
  - nombre d'arêtes incidentes à un sommet donné
  - les boucles comptent deux fois
  - pour les graphes orientés, on distingue degré entrant [in-degree] $\mathbf{d^-}$ et degré sortant [out-degree] $\mathbf{d^+}$
- Les successeurs d'un sommet : $\Gamma\mathbf{(x)}$
- Les prédécesseurs d'un sommet : $\Gamma^{-1}\mathbf{(x)}$

Boucles, sommets

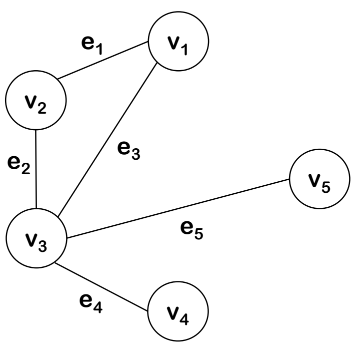
- Une boucle [loop]
  - arête (ou arc) dont les extrémités sont confondues
- Un sommet pendant [pending vertex]
  - sommet qui n'a qu'un seul voisin
- Un sommet isolé
  - sommet qui n'est relié à aucun autre

Chaînes, cycles

- Une chaîne [path] dans **G**
  - suite ayant pour éléments alternativement des sommets et des arêtes, commençant et se terminant par un sommet
  - telle que chaque arête est encadrée par ses extrémités
- La chaîne relie le premier sommet de la suite au dernier sommet
- La chaîne a pour longueur le nombre d'arêtes de la chaîne

Chaînes, cycles

- Le graphe ci-dessous contient entre autres les chaînes $(v_1, e_1, v_2, e_2, v_3, e_5, v_5)$ et $(v_4, e_4, v_3, e_2, v_2, e_1, v_1)$

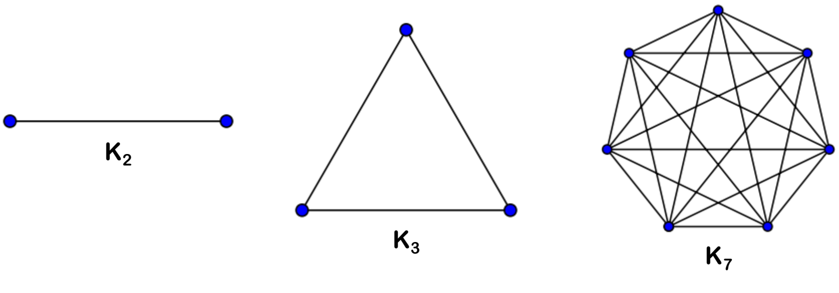


Chaînes, cycles

- distance entre deux sommets
  - longueur de la plus petite chaîne les reliant
- diamètre d'un graphe
  - la plus longue des distances entre deux sommets du graphe
- chaîne élémentaire
  - chaque sommet y apparaît au plus une fois
- chaîne simple
  - chaque arête y apparaît au plus une fois
- chaîne fermée
  - chaîne dont les sommets de départ et de fin sont les mêmes
- cycle [cycle, circuit]
  - chaîne fermée simple

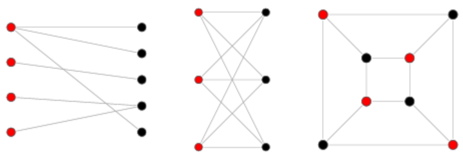
Graphe complet

- Graphe complet [complete graph]
  - graphe dont tous les sommets sont adjacents
  - ont un petit nom : $\mathbf{K_n}$



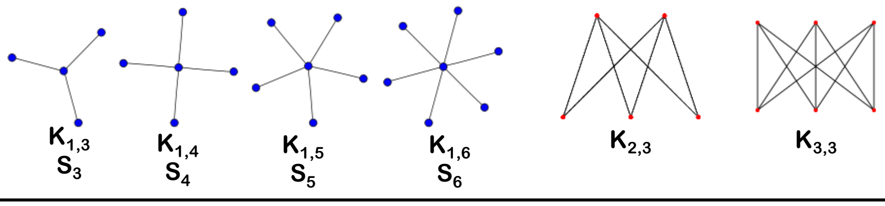
Graphe biparti

- Graphe biparti [bipartite graph]
  - graphe dont l'ensemble des nœuds peut être divisé en deux ensembles disjoints **U** et **V** tels que chaque arête a un sommet en **U** et un sommet en **V**



Graphe biparti complet

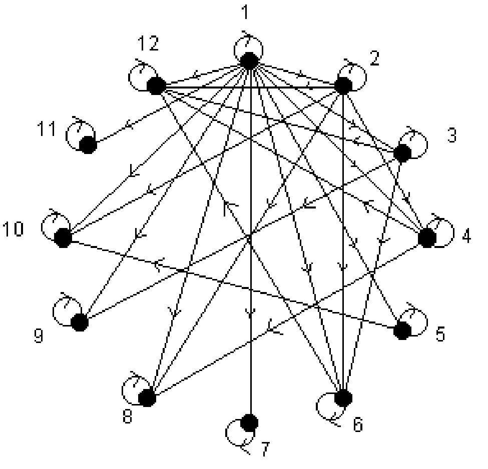
- Graphe biparti complet
  - est biparti
  - contient le nombre maximal d'arêtes
- Il existe une partition de son ensemble de sommets en deux sous-ensembles **U** et **V**
  - chaque sommet de **U** est relié à chaque sommet de **V**
- $\mathbf{K_{n,m}}$



Exercice 1

- Construire un graphe orienté dont les sommets sont les entiers compris entre 1 et 12 et dont les arcs représentent la relation « être diviseur de »

Exercice 1 (correction)



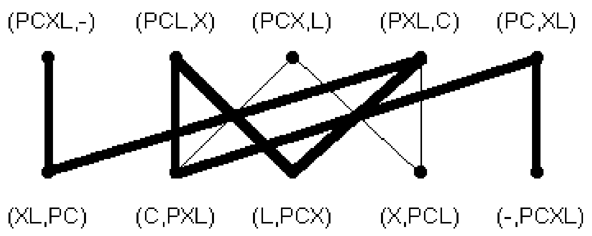
Exercice 2

- Une chèvre, un chou et un loup se trouvent sur la rive d'un fleuve
- Un passeur souhaite les transporter sur l'autre rive mais, sa barque étant trop petite, il ne peut transporter qu'un seul d'entre eux à la fois
- Comment doit-il procéder afin de ne jamais laisser ensemble et sans surveillance ni le loup et la chèvre, ni la chèvre et le chou ?

Exercice 2 (correction)

- Modélisation à l'aide d'un graphe
  - P le passeur
  - C la chèvre
  - X le chou
  - L le loup
  - Sommets du graphe : couples précisant qui est sur la rive de départ, qui est sur la rive d'arrivée
    - exemple : (PCX,L) signifie que le passeur est sur la rive initiale avec la chèvre et le chou (qui sont donc sous surveillance), alors que le loup est sur l'autre rive.
  - Une arête relie deux sommets lorsque le passeur peut passer d'une situation à l'autre
    - exemple : en transportant la chèvre, le passeur passe de (PCX,L) à (X,PCL)
  - Le graphe obtenu est biparti : les sommets pour lesquels le passeur est sur la rive initiale ne sont reliés qu'aux sommets pour lesquels le passeur est sur l'autre rive
  - on ne considère pas les sommets dont l'une des composantes est CX ou LC car ces situations sont interdites
- Il faut alors trouver un chemin entre la situation initiale (PCXL,-) et la situation finale souhaitée (-,PCXL)
  - le plus court chemin nous permet de minimiser le nombre de déplacements

Exercice 2 (correction)



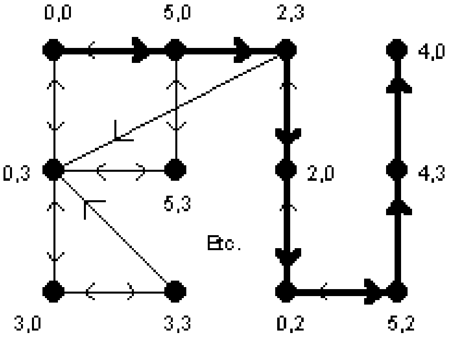
Exercice 3

- On souhaite prélever 4 litres de liquide dans un tonneau. Pour cela, nous avons à notre disposition deux récipients (non gradués !), l'un de 5 litres, l'autre de 3 litres... Comment doit-on faire ?

Exercice 3 (correction)

- Même principe que pour l'exercice 2, modélisation sous forme de graphe
  - les sommets sont cette fois des couples donnant le contenu du récipient de 5 litres et celui du récipient de 3 litres
  - un arc entre deux sommets signifie qu'on peut passer d'une situation à l'autre
  - on cherche un chemin du sommet 0,0 au sommet 4,0

Exercice 3 (correction)



Exercice 4

- Soit **X** un ensemble de lapins, et **G** un graphe orienté ayant **X** pour ensemble de sommets.
- **G** est un « graphe de parenté » si les arcs de **G** codent la relation « être l'enfant de »
- Quelles conditions doit nécessairement vérifier **G** pour pouvoir être un graphe de parenté ?

Exercice 4 (correction)

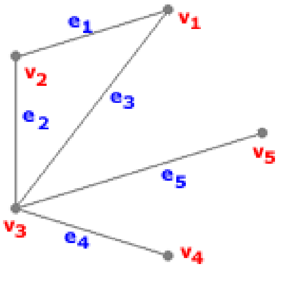
- Voici une liste de conditions nécessaires :
  - chaque sommet doit avoir un degré entrant égal à 2
    - chaque lapin a deux parents
    - à l'exception de deux sommets pour lesquels le degré entrant est nul (« Adam » et « Ève »)
  - le graphe doit être sans circuit
    - un lapin ne peut avoir pour parent l'un de ses descendants
  - on doit pouvoir colorier les sommets de ce graphe en deux couleurs
    - male et femelle
    - de façon telle que tout sommet de degré entrant égal à 2 possède un prédécesseur male et un prédécesseur femelle

Graphe partiel

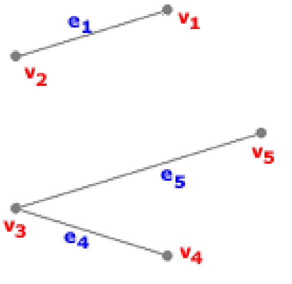
- Soit **G = (V, E)** un graphe
- Le graphe **G' = (V, E')** est un graphe partiel de **G**, si **E'** est inclus dans **E**
- Autrement dit, on obtient **G'** en enlevant une ou plusieurs arêtes au graphe **G**

Graphe partiel

G=(V, E)
$V=\{v_1, v_2, v_3, v_4, v_5\}$
$E=\{e_1=(v_1,v_2)$,
$e_2=(v_2,v_3)$, $e_3=(v_1,v_3)$,
$e_4=(v_3,v_4)$, $e_5=(v_3,v_5)\}$



G'=(V',E')
V'=V
E'={e1, e4, e5}

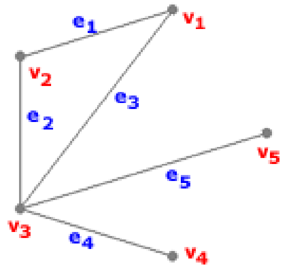


Sous-graphe

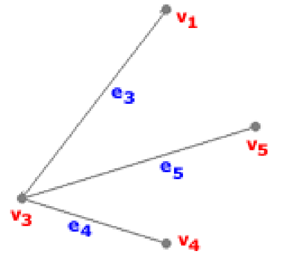
- Soit **G = (V, E)** un graphe
- Pour un sous-ensemble de sommets **A** inclus dans **V**, le sous-graphe de **G** induit par **A** est le graphe **G' = (A, E(A))** dont l'ensemble des sommets est A et l'ensemble des arêtes **E(A)** est formé de toutes les arêtes de **G** ayant leurs deux extrémités dans **A**
- Autrement dit, on obtient **G'** en enlevant un ou plusieurs sommets au graphe **G**, ainsi que toutes les arêtes incidentes à ces sommets

Sous-graphe

G=(V, E)
V={v_1, v_2, v_3, v_4, v_5}
E={e_1=(v_1,v_2), e_2=(v_2,v_3), e_3=(v_1,v_3), e_4=(v_3,v_4), e_5=(v_3,v_5)}



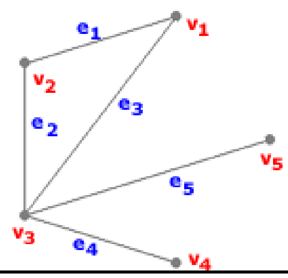
G'=(V',E')
V'={v1, v3, v4, v5}
E'={e3, e4, e5}



Sous-graphe partiel

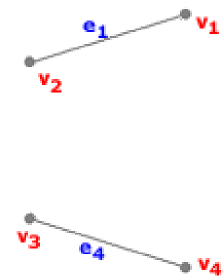
- Un graphe partiel d'un sous-graphe est un sous-graphe partiel de G

G=(V, E)
V={v_1, v_2, v_3, v_4, v_5}
E={e_1=(v_1,v_2), e_2=(v_2,v_3), e_3=(v_1,v_3), e_4=(v_3,v_4), e_5=(v_3,v_5)}



G'=(V',E')

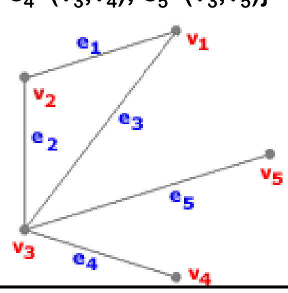
V'={v1, v2, v3, v4}
E'={e1, e4}



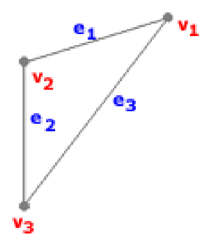
Clique

- On appelle clique un sous-graphe complet de G

G=(V, E)
V={v_1, v_2, v_3, v_4, v_5}
E={e_1=(v_1,v_2), e_2=(v_2,v_3), e_3=(v_1,v_3), e_4=(v_3,v_4), e_5=(v_3,v_5)}



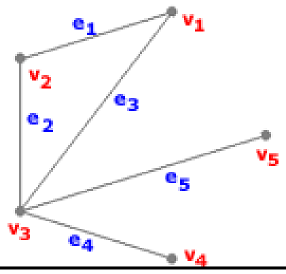
V'={v1,v2,v3}
E'={e1, e2, e3}
est une clique de **G**



Stable

- On appelle stable un sous-graphe de G sans arêtes

G=(V, E)
V={v_1, v_2, v_3, v_4, v_5}
E={e_1=(v_1,v_2), e_2=(v_2,v_3),
e_3=(v_1,v_3),
e_4=(v_3,v_4), e_5=(v_3,v_5)}



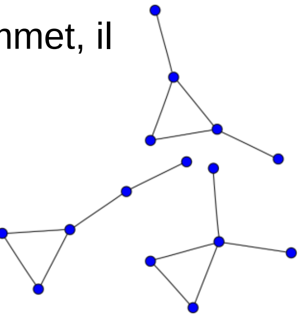
V'={v_1,v_4,v_5}
E'={}
est un stable de **G**

Connexité

- Un graphe est connexe [connected] s'il est possible, à partir de n'importe quel sommet, de rejoindre tous les autres en suivant les arêtes
- Le réseau du métro parisien est connexe

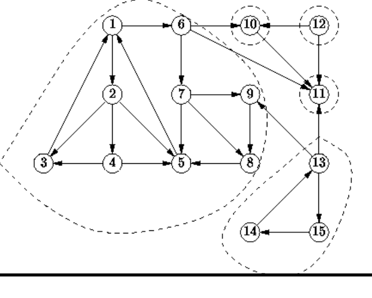
Composante connexe

- Un sous-graphe connexe maximal d'un graphe non orienté est une composante connexe [connected component] de ce graphe
- Le sous-graphe connexe est maximal : si on ajoutait un sommet, il ne serait plus connexe
- Un graphe non connexe se décompose en composantes connexes



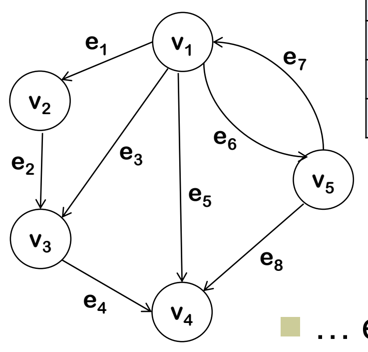
Composante fortement connexe

- Une composante fortement connexe [strongly connected component] d'un graphe orienté **G** est un sous-graphe de **G** tel que
  - pour tout couple **(u, v)** de nœuds dans ce sous-graphe il existe un chemin de **u** à **v**
  - ce sous-graphe est maximal



Représentation des graphes en machine

- Listes de successeurs

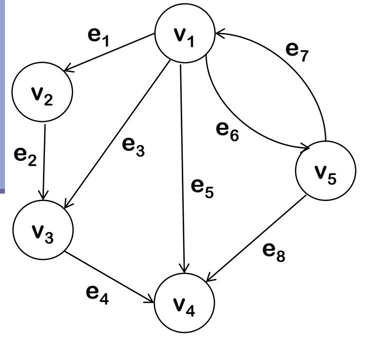


v_1	v_2	v_3	v_4	v_5
v_2	v_3			
v_3	v_4			
v_4				
v_5	v_1	v_4		

- … et/ou listes de prédécesseurs

Représentation des graphes en machine

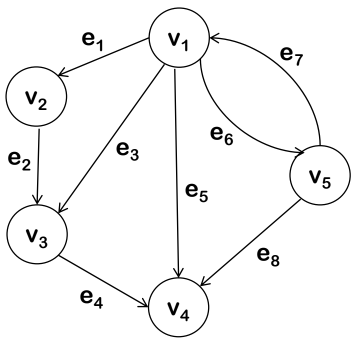
- Matrice d'adjacence
  - $u_{i,j}$ **= 1** si **(i,j)** est un arc, **0** sinon



	v_1	v_2	v_3	v_4	v_5	
v_1	0	1	1	1	1	$d^+(v1)$
v_2	0	0	1	0	0	
v_3	0	0	0	1	0	
v_4	0	0	0	0	0	
v_5	1	0	0	1	0	
	$d^-(v1)$					

Représentation des graphes en machine

- Matrice d'incidence
  - $\mathbf{u_{i,j} = -1}$ si $\mathbf{v_i}$ est origine de $\mathbf{e_j}$
  - $\mathbf{u_{i,j} = 1}$ si $\mathbf{v_i}$ est extrémité de $\mathbf{e_j}$
  - **0** sinon



	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
v_1	-1	0	-1	0	-1	-1	1	0
v_2	1	-1	0	0	0	0	0	0
v_3	0	1	1	-1	0	0	0	0
v_4	0	0	0	1	1	0	0	1
v_5	0	0	0	0	0	1	-1	-1

Exercice 5

- Pour chacune des représentations, évaluer la complexité algorithmique des opérations suivantes
  - est-ce que $\mathbf{v_i}$ et $\mathbf{v_j}$ sont voisins ?
  - combien $\mathbf{v_i}$ a-t-il de voisins ?
  - quels sont les voisins de $\mathbf{v_i}$?
  - supprimer l'arête $\mathbf{(v_i,v_j)}$
  - supprimer le premier voisin de $\mathbf{v_i}$

TD 1

- Dans le langage de votre choix, proposez une implémentation de l'objet mathématique graphe permettant notamment
  - de créer et détruire des sommets, des arcs entre sommets
  - à partir d'un sommet, d'accéder aux arcs incidents et aux sommets voisins
- Ecrivez une fonction (récursive ?) qui cherche un chemin entre deux sommets
- Ecrivez une fonction qui vérifie la connexité d'un graphe