Théorie des graphes

Michel Rigo

http://www.discmath.ulg.ac.be/

Année 2019–2020

Organisation de l'année

- Cours théorique
- Séances d'exercices

Notes de cours, journal de bord, ...

- `http://www.discmath.ulg.ac.be/`

Podcasts via `MyULg`

Deux modes d'examen possibles

1. - Examen écrit d'exercices
   - Examen écrit portant sur la théorie (± 25 questions)
2. - Projet de programmation (rapport, présentation orale)
   - Examen écrit exercices + théorie "élémentaire"

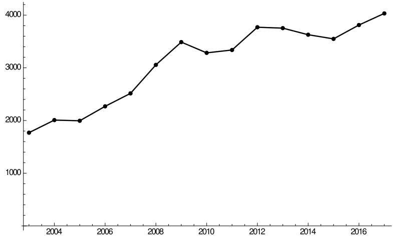
Plan du cours

- premières notions, graphes orientés/non orientés
- chemins, graphes eulériens
- recherche du plus court chemin
- tri topologique
- arbres
- graphes hamiltoniens
- théorie algébrique des graphes (PREREQUIS)
- application au PageRank
- planarité, formule d'Euler
- coloriages, théorème des 4 couleurs, nombres de Ramsey

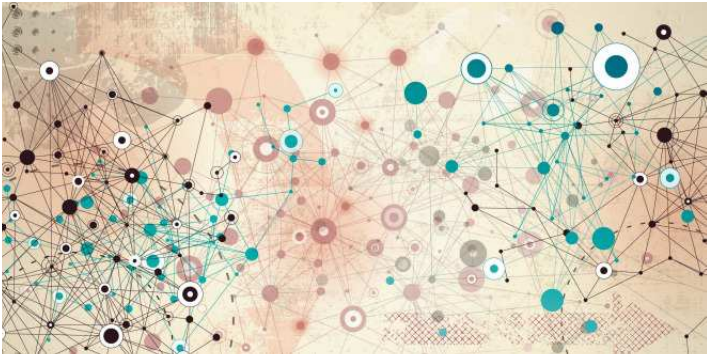
Pas d'analyse de complexité d'algorithmes (P vs NP)

Quelques repères

- sept ponts de Königsberg étudié (Euler, 1736)
- "*Theorie der endlichen und unendlichen Graphen*" (König, 1936)
- milieu du vingtième siècle
  N. Biggs, C. Berge, P. Erdős, W.T. Tutte, ...
- 2010 Math. Subject Classification `05C` Graph theory
  - En 2016, 3812 articles de recherche "`05C`"
  - En 2017, 4032 articles de recherche "`05C`"



Is Graph Theory Key to Understanding Big Data ?
WIRED Mars 2014



iStock.com/AF-studio

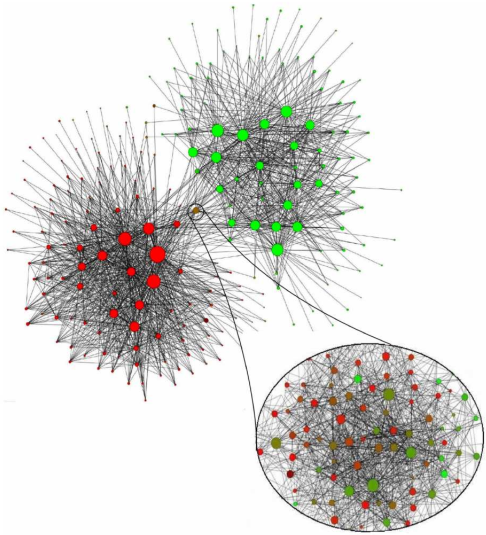
Graph Theory : Key to Understanding Big Data
WIRED Mai 2014

Facebook CEO Mark Zuckerberg shows off Graph Search.

Photo : Ariel Zambelich/Wired

http ://innovationinsights.wired.com/insights/2014/05/graph-theory-key-understanding-big-data-2/

Les graphes 'réels' sont 'grands'



Réseau Mobistar $\pm 2.10^6$ abonnés – arXiv0803.0476, V. Blondel et al.

graphe **orienté**

DÉFINITION

- V un ensemble (fini ou infini)
- E une partie de $V \times V$ (i.e., une relation sur V).

Le graphe $G = (V, E)$ est la donnée du couple (V, E).

Les éléments de V sont les sommets de G (vertex/vertices).

Les éléments de E sont les arcs de G (edges).

Si V est fini, on parlera de graphe fini et $\#E \leq (\#V)^2$.

⚠ ordre au sein des couples appartenant à E

couple $(x, y) \neq$ paire $\{x, y\}$

GRAPHE ORIENTÉ - VOCABULAIRE

Soient $V = \{v_i \mid i \in I\}$ et $a = (v_i, v_j)$, $i, j \in I$

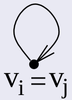
l'origine v_i et la destination v_j de l'arc a.

v_i et v_j sont les extrémités de l'arc a

a relie v_i à v_j.

Si $b = (v_i, v_i)$: b est une boucle.

Deux arcs adjacents ont au moins une extrémité en commun.

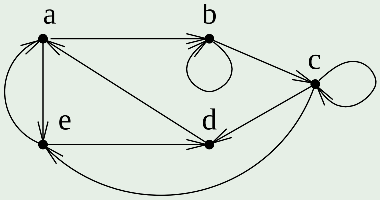


Un graphe orienté

Soit le graphe $G = (V, E)$ où $V = \{a, b, c, d, e\}$ et

$$E = \{(a, b), (a, e), (b, b), (b, c), (c, c), (c, d),$$

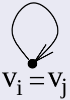
$$(c, e), (d, a), (e, a), (e, d)\}.$$



Il s'agit d'UNE *représentation* du graphe (parmi d'autres)

Graphe orienté - vocabulaire

Soit $a = (v_i, v_j) \in E$.



a est un arc sortant de v_i ou arc incident à v_i vers l'extérieur

a est un arc entrant dans v_j ou arc incident à v_j vers l'intérieur

ensemble des arcs sortant de v_i $= \omega^+(v_i)$

ensemble des arcs entrant dans v_j $= \omega^-(v_j)$

ensemble des arcs incidents $= \omega(v) := \omega^+(v) \cup \omega^-(v)$

demi-degré sortant (resp. demi-degré entrant) d'un sommet v

$$d^+(v) = \#(\omega^+(v)) \quad d^-(v) = \#(\omega^-(v)).$$

Handshaking formula

$$\sum_{v \in V} d^+(v) = \sum_{v \in V} d^-(v).$$

Définition (suite. . .)

degré de v : $\deg(v) = d^+(v) + d^-(v)$

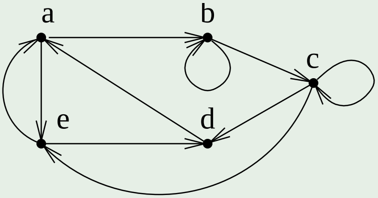
ensemble des successeurs de v : $\mathrm{succ}(v) = \{s_1, \ldots, s_k\}$
sommets s_i tels que $(v, s_i) \in \omega^+(v)$, i.e., $(v, s_i) \in E$.

ensemble des prédécesseurs de v : $\mathrm{pred}(v) = \{s_1, \ldots, s_k\}$
sommets s_i tels que $(s_i, v) \in \omega^-(v)$, i.e., $(s_i, v) \in E$.

ensemble des voisins de v : $\nu(v) = \mathrm{pred}(v) \cup \mathrm{succ}(v)$

Si u appartient à $\nu(v)$, u et v sont des sommets voisins

Exemple (suite...)



$\omega^+(a) = \{(a,b),(a,e)\}$, $\omega^-(d) = \{(c,d),(e,d)\}$,

$\mathrm{succ}(a) = \{b,e\}$, $\mathrm{succ}(b) = \{b,c\}$, $\mathrm{pred}(d) = \{c,e\}$,

$\nu(a) = \{b,d,e\}$,

les arcs (e,a) et (d,a) sont adjacents,

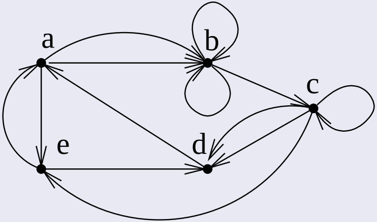
$d^+(c) = 3$

Définition naïve

Un multi-ensemble : $\{1, 1, 2, 3\}$, $\{1, 2, 3\}$ et $\{1, 2, 2, 3\}$
$\{1_1, 1_2, 1_3, 2_1, 2_2, 3\}$

Définition

Un multi-graphe $G = (V, E)$ est un graphe pour lequel l'ensemble E des arcs est un multi-ensemble.



Un multi-graphe $G = (V, E)$ est fini si V et E sont finis

⚠ V fini n'implique pas E fini.

Définition (suite. . .)

Soit $p \geq 1$. Un p-graphe est un multi-graphe $G = (V, E)$ pour lequel tout arc de E est répété au plus p fois.

En particulier, un 1-graphe est un graphe.

Pour les multi-graphes

"handshaking formula" OK

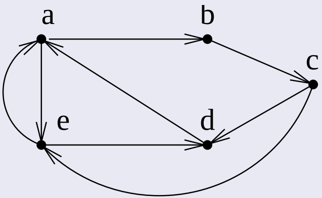
on adapte $\omega^+(v)$, $d^+(v)$, $\mathrm{succ}(v)$ et $\omega^-(v)$, $d^-(v)$, $\mathrm{pred}(v)$.

$\omega^+(v)$ et $\omega^-(v)$ sont en général des multi-ensembles.

Définition

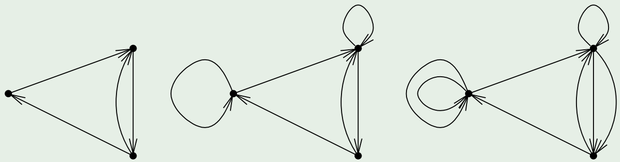
Un graphe $G = (V, E)$ est simple

- il ne s'agit pas d'un multi-graphe, pas d'arête multiple,
- E est irréflexif : $\forall v \in V$, $(v, v) \notin E$, pas de boucle



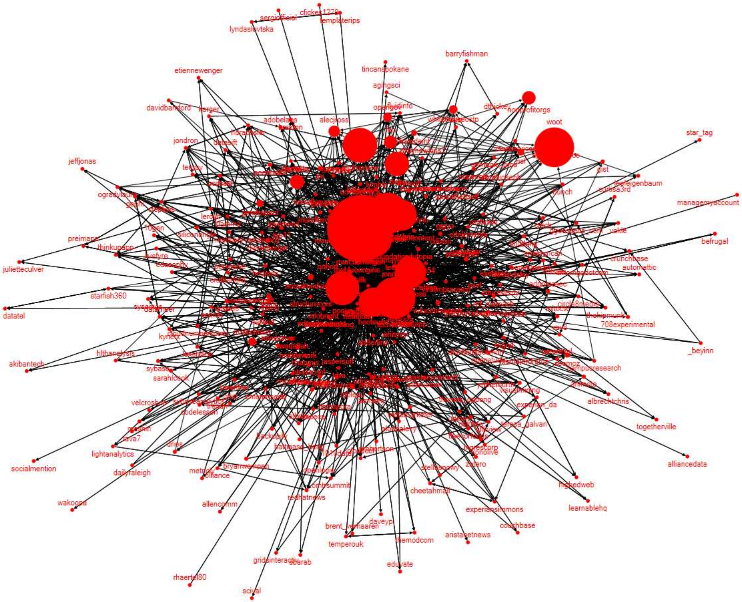
En résumé

Un graphe (orienté) simple, un graphe et un multi-graphe



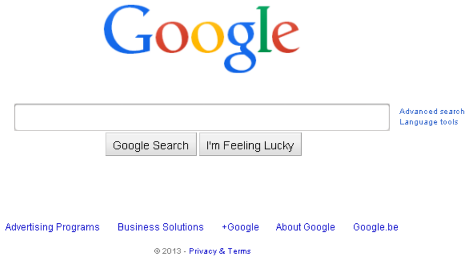
Quelques exemples de graphes orientés

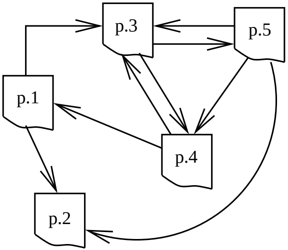
- le graphe de Twitter
- le graphe des pages web
- les flux de migration entre pays du globe
- un réseau routier avec des sens uniques
- relation d'ordre
- graphe de Cayley d'un groupe
- graphe de De Bruijn



Visualizing my Twitter Network by number of followers.

Michael Atkisson http ://woknowing.wordpress.com/



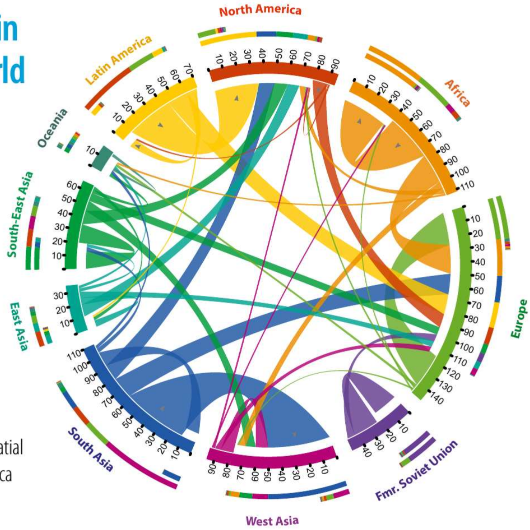


Les pages et les liens entre les pages

Migration flows within and between ten world regions, in 100,000's

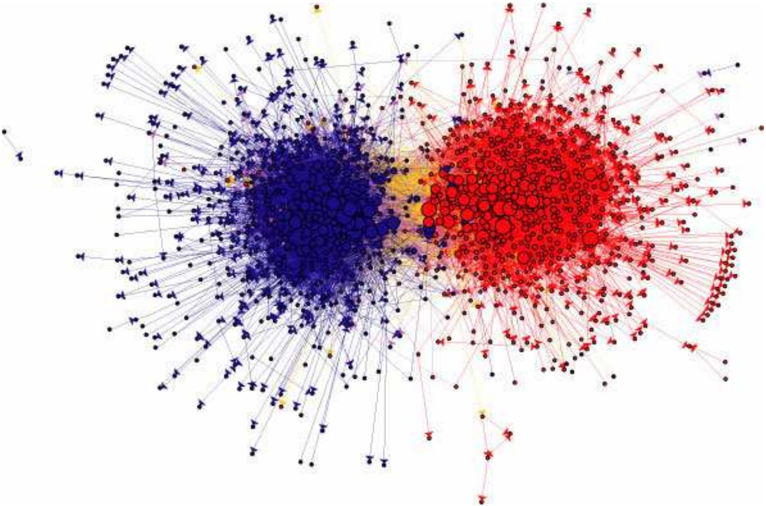
This circular plot shows all global bilateral migration flows for the five-year period mid-2005 to mid-2010, classified into a manageable set of ten world regions.

Key features of the global migration system include the high concentration of African migration within the continent (with the exception of Northern Africa), the 'closed' migration system of the former Soviet Union, and the high spatial focus of Asian emigration to North America and the Gulf states.

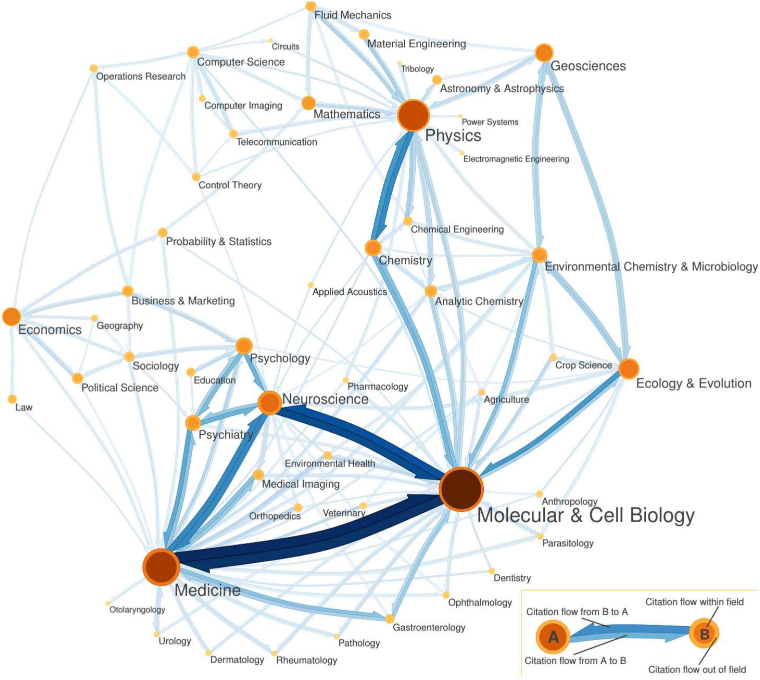


http://download.gsb.bund.de/BIB/global_flow/

Quantifying Global International Migration Flows, G. J. Abel, N. Sande, Science 2014 blogs politiques avant l'élection présidentielle aux USA de 2004.



http ://www-personal.umich.edu/~ladamic/img/politicalblogs.jpg



http://eigenfactor.org

Une application en théorie des groupes

Définition (Graphe de Cayley)

Soit $\mathbb{G}$ un groupe et S un ensemble de générateurs de $\mathbb{G}$.

Tout élément de $\mathbb{G}$ s'obtient comme produit d'un nombre fini d'éléments de S ou d'inverses d'éléments de S.

Le graphe de Cayley du groupe $\mathbb{G}$ (par rapport à S) est un graphe orienté $\mathcal{C}_S(\mathbb{G})$ ayant pour sommets les éléments de $\mathbb{G}$.

Pour tous $g \in \mathbb{G}$, $s \in S$, l'arc (g, gs) est un arc de label s.

Remarque : un tel graphe peut être infini.

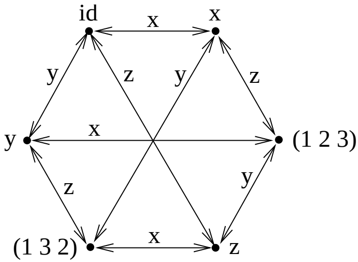
Graphe de Cayley

$\mathcal{S}_3, \quad S = \{x = \begin{pmatrix}1 & 2\end{pmatrix}, y = \begin{pmatrix}1 & 3\end{pmatrix}, z = \begin{pmatrix}2 & 3\end{pmatrix}\}$

$$xx = yy = zz = id, \quad xy = \begin{pmatrix}1 & 2\end{pmatrix}\begin{pmatrix}1 & 3\end{pmatrix} = \begin{pmatrix}1 & 3 & 2\end{pmatrix},$$

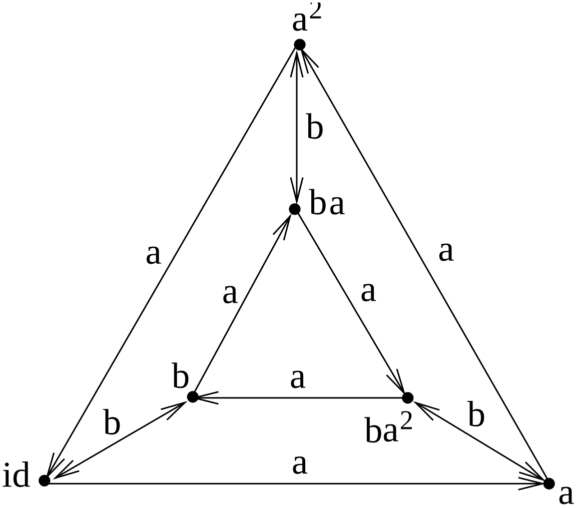
$$xz = \begin{pmatrix}1 & 2\end{pmatrix}\begin{pmatrix}2 & 3\end{pmatrix} = \begin{pmatrix}1 & 2 & 3\end{pmatrix}, \quad yz = \begin{pmatrix}1 & 3\end{pmatrix}\begin{pmatrix}2 & 3\end{pmatrix} = \begin{pmatrix}1 & 3 & 2\end{pmatrix}$$

$$yx = \begin{pmatrix}1 & 2 & 3\end{pmatrix}, \; zx = \begin{pmatrix}1 & 3 & 2\end{pmatrix}, \; zy = \begin{pmatrix}1 & 2 & 3\end{pmatrix}.$$

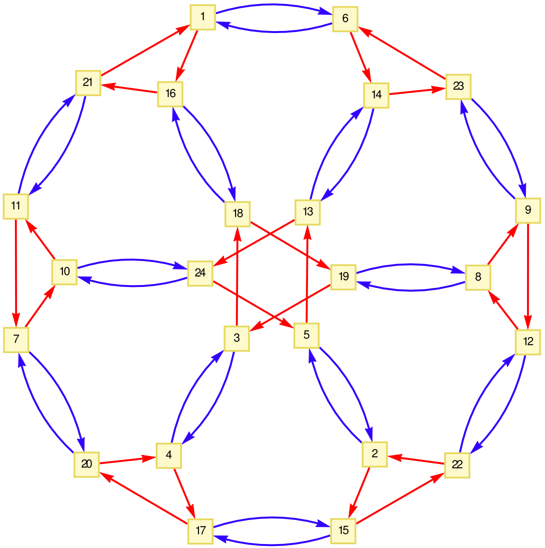


Graphe de Cayley

$\mathcal{S}_3, \quad S = \{a = \begin{pmatrix}1 & 2 & 3\end{pmatrix}, b = \begin{pmatrix}1 & 2\end{pmatrix}\}$

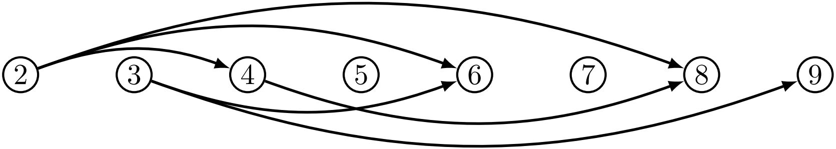


Dans `Mathematica` : `CayleyGraph[]`



$(1\ 2\ 4)$ et $(3\ 4)$

relation de divisibilité



Exemple de graphe infini $V = \{2, 3, 4, \ldots\}$

Définition

Soit $G = (V, E)$ un graphe (resp. un multi-graphe).

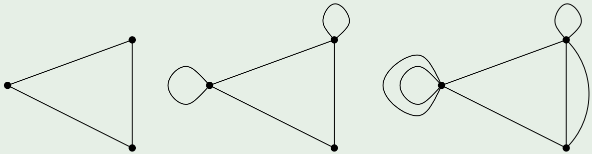
Si E est une relation symétrique sur V,
alors G est un graphe (resp. un multi-graphe) non orienté, i.e.,

$$\forall v_1, v_2 \in V : (v_1, v_2) \in E \Rightarrow (v_2, v_1) \in E.$$

On identifie les arcs (v_i, v_j) et (v_i, v_j)
avec une unique arête (non orientée) : la paire $\{v_i, v_j\}$.

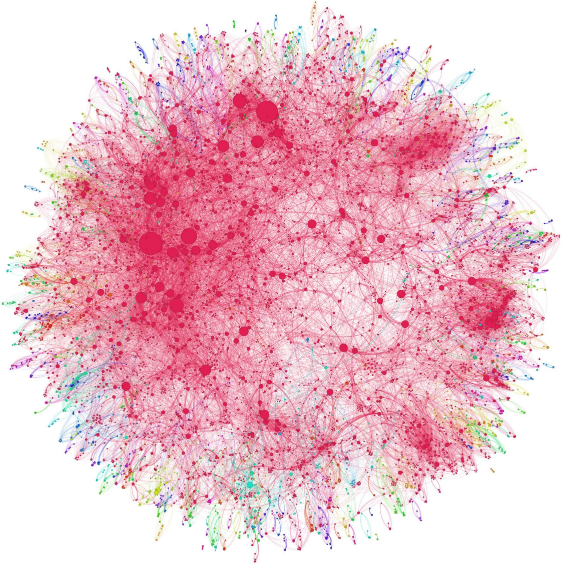
En résumé

Un graphe (non orienté) simple, un graphe et un multi-graphe



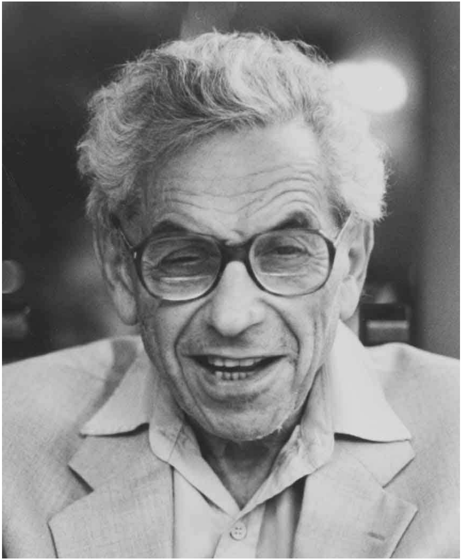

facebook

Des co-auteurs scientifiques et le nombre d'Erdős

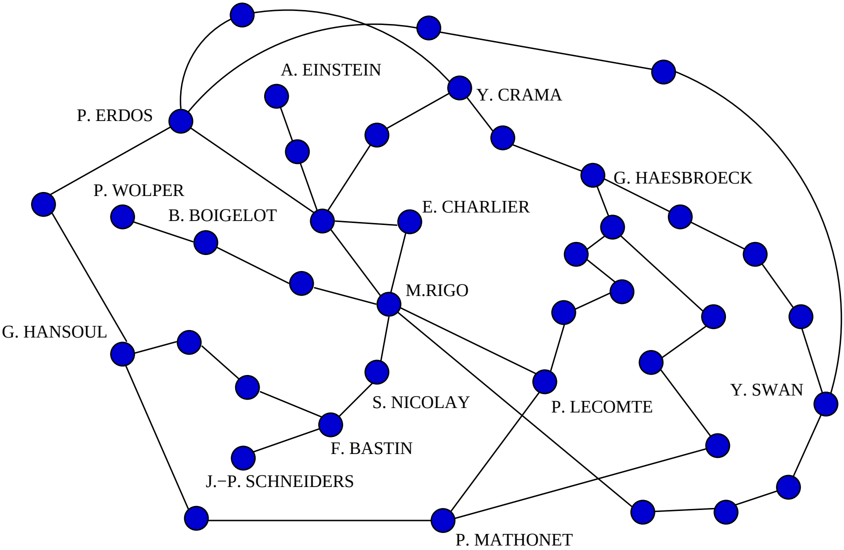


co-authorship network of 8,500 doctors and scientists publishing on hepatitis C virus between 2008 and 2012

M. Newman, the structure of scientific collaboration networks (2001).



Paul Erdős (1913–1996), mathématicien hongrois
Total Publications : 1425, Total Citations : 12847

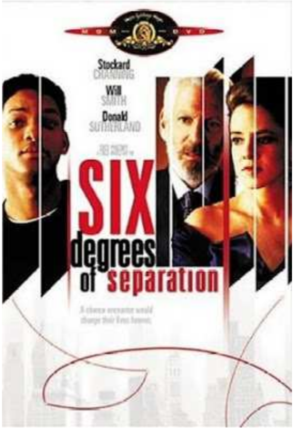


sous-graphe, chemins les plus courts (septembre 2014)

M. Newman, the structure of scientific collaboration networks (2001).

Small-world phenomenom

"I read somewhere that everybody on this planet is separated only by six other people. *Six degrees of separation*. Between us and everyone else on this planet."



John Guare

Quel est votre nombre de ? ? ?

Un peu de vocabulaire

Soient $G = (V, E)$ multi-graphe **non orienté**, $a = \{v_i, v_j\} \in E$.

a est incident à v_i et v_j.

degré de v_i, $\deg(v_i) = \#$ arêtes incidentes à v_i.

⚠ les **boucles** apportent une double contribution au degré.

ensemble des arêtes incidentes à v_i : $\omega(v_i)$.
Si G est simple, $\deg(v_i) = \#(\omega(v_i))$.

$\leadsto$ Notations compatibles avec le cas orienté.

Deux arêtes sont adjacentes si au moins une extrémité en commun.

Deux sommets $v_i, v_j \in V$ sont adjacents/voisins, si $\{v_i, v_j\} \in E$.

ensemble des voisins de v : $\nu(v)$

p-graphe : analogue avec cas cas orienté.

Handshaking formula (bis)

Si $G = (V, E)$ est un multi-graphe non orienté, alors

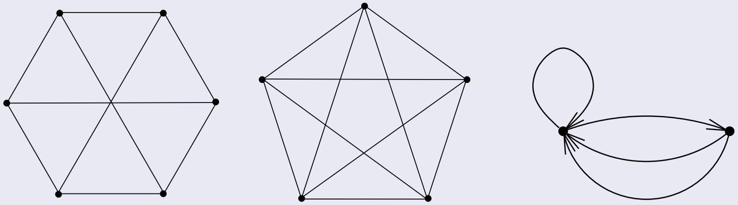
$$\sum_{v \in V} \deg(v) = 2\,\#E.$$

On comprend mieux la double contribution des boucles pour le degré d'un sommet...

Définition

Soit $k \geq 1$. Un multi-graphe orienté (resp. non orienté) $G = (V, E)$ est *k-régulier* si

$$\forall v \in V,\ d^+(v) = k \quad \text{(resp. } \deg(v) = k\text{)}.$$

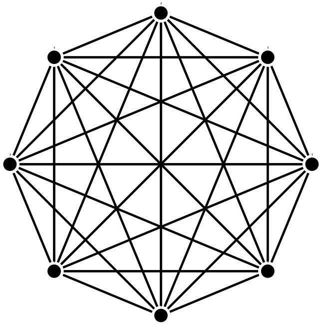


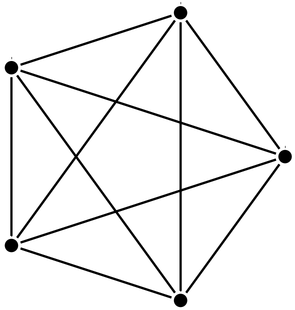
Définition (suite...)

Un graphe $G = (V, E)$ est complet si $E = V \times V$ et il est sous-entendu que ce graphe est simple et non orienté, i.e.,

$$E = V \times V \setminus \{(v, v) \mid v \in V\}$$

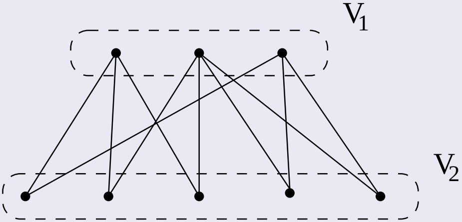
notation K_n





Définition

Un graphe $G = (V, E)$ est biparti
si V partitionné en V_1 et V_2 t.q. $E \subseteq V_1 \times V_2$.

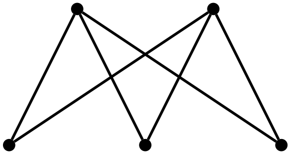


Graphe biparti complet $K_{m,n}$:

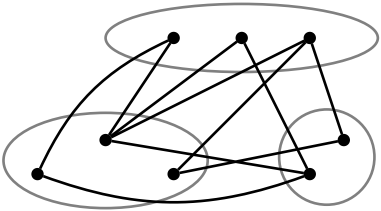
$$\# V_1 = m, \quad \# V_2 = n, \quad E = V_1 \times V_2.$$

graphes n-partis, $n \geq 2$,
V partitionné en n sous-ensembles $V_1, \ldots, V_n$ t.q.

$$E \subseteq \bigcup_{i \neq j} V_i \times V_j .$$

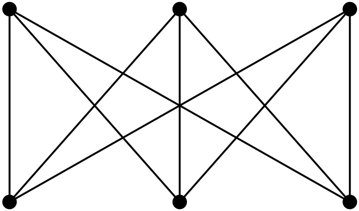


$K_{2,3}$



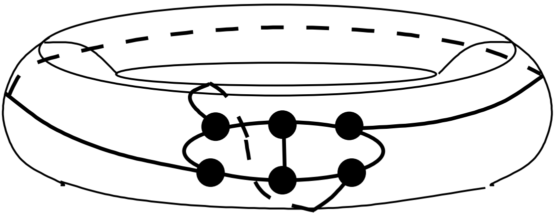
Le *problèmes des 3 villas*, eau, gaz, électricité,. . .

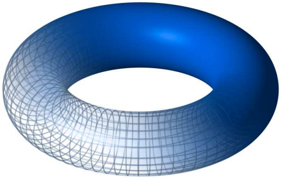
$K_{3,3}$



Démonstration ? nous verrons que $K_{3,3}$ n'est pas planaire. . .

Sur un tore (surface de genre 1)





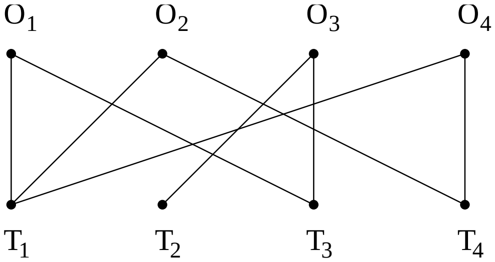
Problèmes d'affectation

ouvriers : $O_1, \ldots, O_k$

postes de travail : $T_1, \ldots, T_t$

Chaque ouvrier O_i possède certaines qualifications lui permettant de travailler sur certains postes $T_{i,1}, \ldots, T_{i,d_i}$.

Comment répartir les ouvriers pour que chaque poste de travail soit occupé par au moins un ouvrier ?



Définition

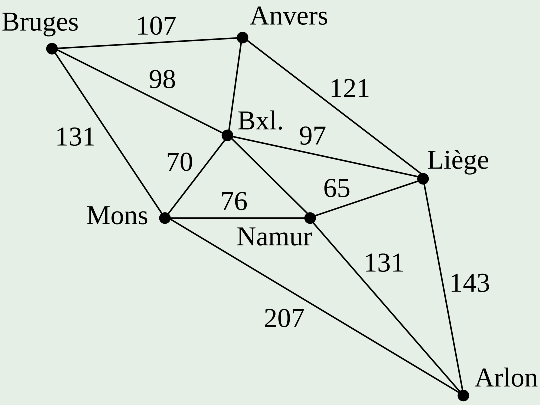
Un multi-graphe $G = (V, E)$ (orienté ou non) est étiqueté (par f) s'il existe une fonction
$$f : E \to \Sigma$$

Si $\Sigma \subseteq \mathbb{R}^+ = [0, +\infty[$, on parle de multi-graphe pondéré, f est une fonction de poids.

Idem avec $g : V \to \Sigma$.

par exemple, une fonction de coloriage

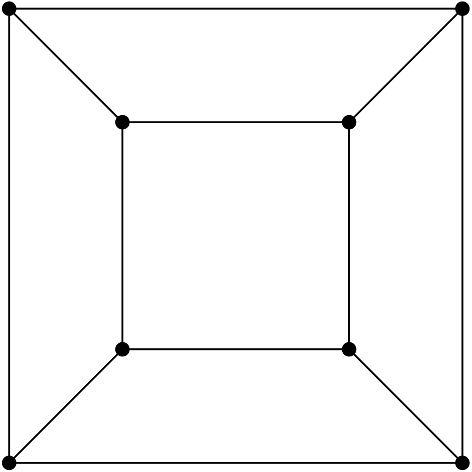
Exemple, autoroutes belges



recherche d'un plus court chemin, algorithme de Dijkstra

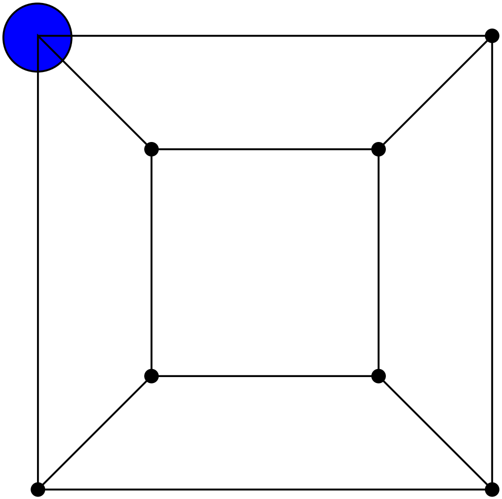
Coloriage

Colorier les sommets d'un cube. . . (coloriage propre)



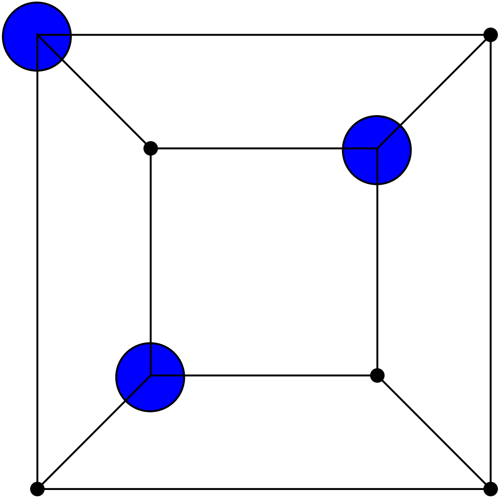
Coloriage

Colorier les sommets d'un cube... (coloriage propre)



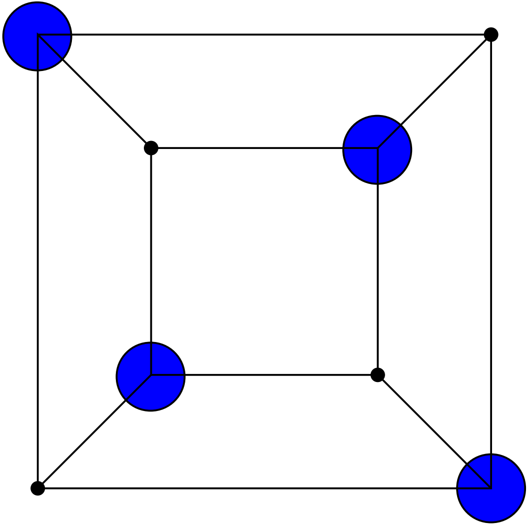
Coloriage

Colorier les sommets d'un cube. . . (coloriage propre)



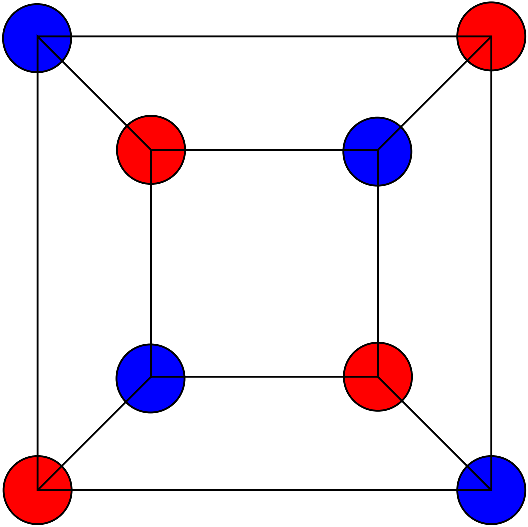
Coloriage

Colorier les sommets d'un cube. . . (coloriage propre)



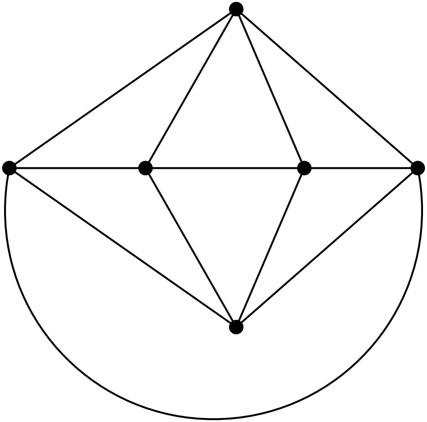
Coloriage

Colorier les sommets d'un cube. . . (coloriage propre)



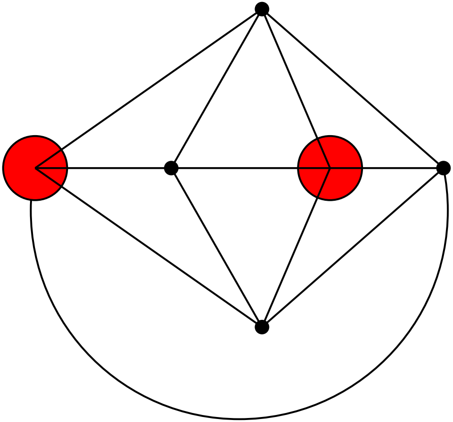
Coloriage

Colorier les faces d'un cube. . . (dual / coloriage propre)



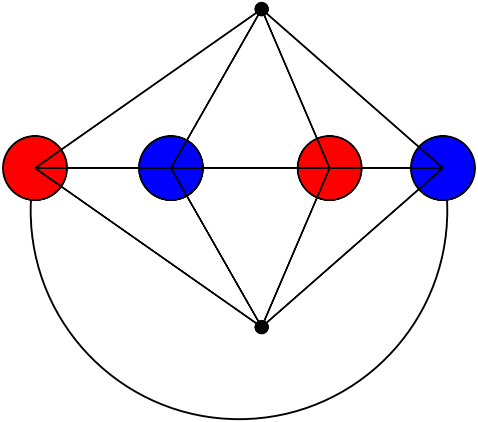
Coloriage

Colorier les faces d'un cube. . . (dual / coloriage propre)



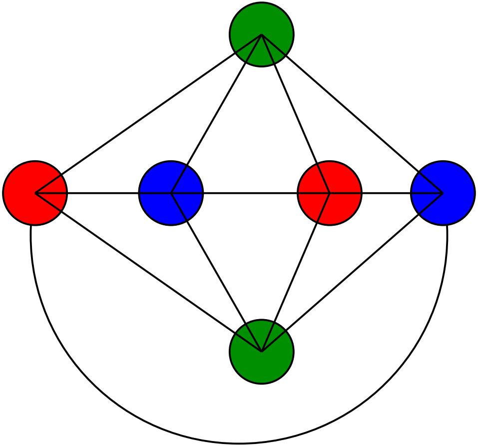
Coloriage

Colorier les faces d'un cube... (dual / coloriage propre)

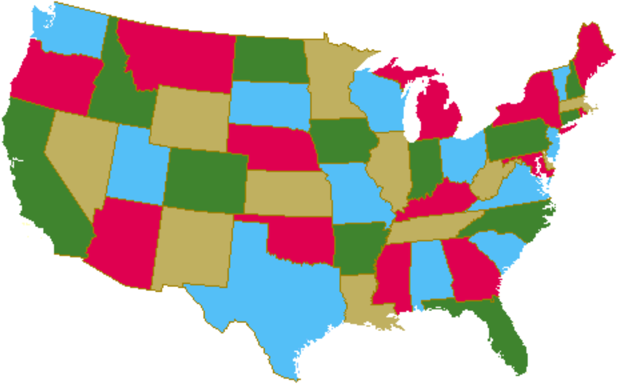


Coloriage

Colorier les faces d'un cube. . . (dual / coloriage propre)

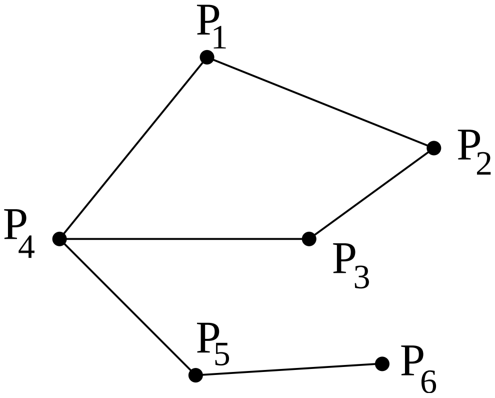


Théorème des 4 couleurs. . .

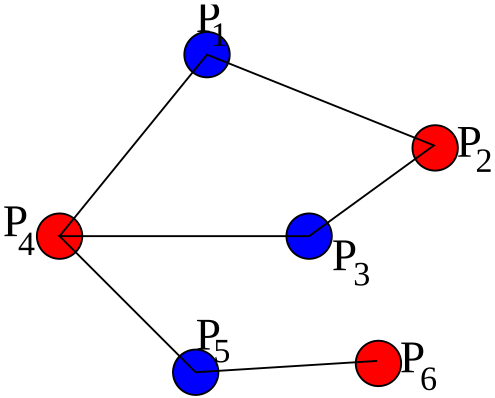


Coloriage

Graphe d'incompatibilité, transport de produits chimiques par wagons — minimiser le nombre de wagons / assurer la sécurité

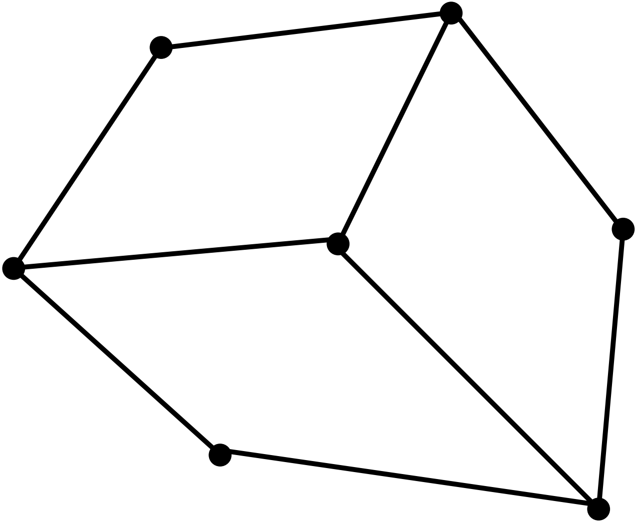


Graphe d'incompatibilité, transport de produits chimiques par wagon



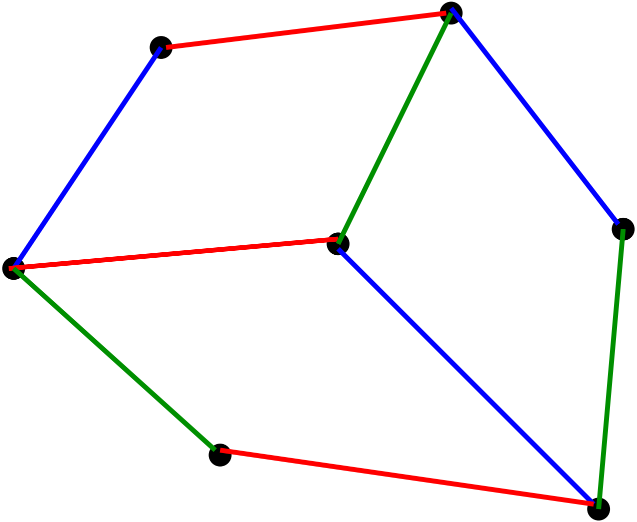
Coloriage

Coloriage d'arêtes… $g : E \to \Sigma$



Coloriage

Coloriage d'arêtes. . .



Quelques notions complémentaires

Définition

Un hyper-graphe $H = (V, E)$.

V =ensemble des sommets de H

E est une partie de $\mathcal{P}(V)$.

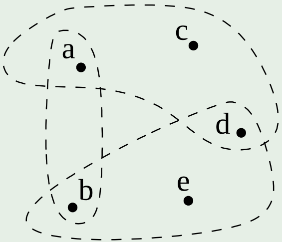
Un élément de E est appelé hyper-arête.

Un hyper-graphe $H = (V, E)$ est fini si V est fini.

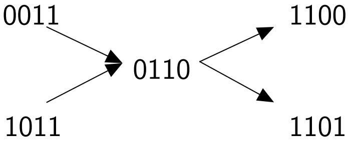
Exemple d'hyper-graphe

Soient $V = \{a, b, c, d, e\}$ et

$$E = \{\{a, b\}, \{a, c, d\}, \{b, d, e\}\}.$$



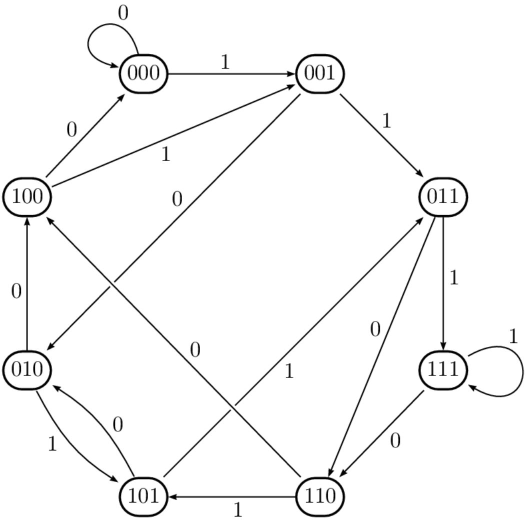
On va définir le graphe de De Bruijn[1] d'ordre n sur l'alphabet A
$\#V = (\#A)^n$, graphe $(\#A)$-régulier



- exemple de graphe $(\#A)$-régulier
- exemple de graphe Eulérien
- exemple de graphe Hamiltonien
- tour de magie

1. Nicolaas Govert de Bruijn (1918–2012)

Graphe de De Bruijn (d'ordre 3 sur deux symboles)



Combinatoire des mots

mots infini = suite infinie $w : \mathbb{N} \to \Sigma$

$$w = abbabaabbaababba\cdots$$

Un *sous-graphe* du graphe de De Bruijn

Définition

Le graphe de Rauzy[2] d'ordre k du mot w,

$V = \{w_i \cdots w_{i+k-1} \mid i \geq 0\}$, i.e. facteurs de longueur k de w

arc de label $\sigma \in \Sigma$ entre les sommets τx et $x\sigma$
SSI $\tau x \sigma$ est un facteur de longueur $k+1$ de w.

2. Gérard Rauzy (1928–2010), théorie ergodique

$ababab ab \cdots$.

$$\begin{tikzcd} \boxed{\text{aba}} \arrow[r, bend left, "\text{b}"] & \boxed{\text{bab}} \arrow[l, bend left, "\text{a}"] \end{tikzcd}$$

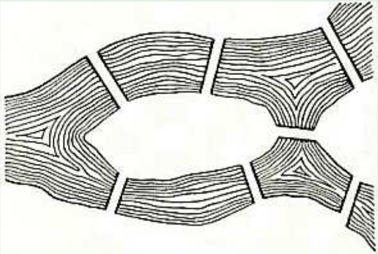
Théorème

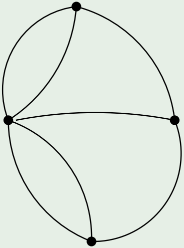
w est ultimement périodique ($w = uvvv\cdots$) SSI,
pour tout k suffisamment grand,
son graphe de Rauzy d'ordre k contient un unique cycle dont tous les sommets ont un demi-degré sortant égal à 1.

Graphes eulériens,
Chemins & Connexité

LES SEPT PONTS DE KÖNIGSBERG / CIRCUIT EULÉRIEN

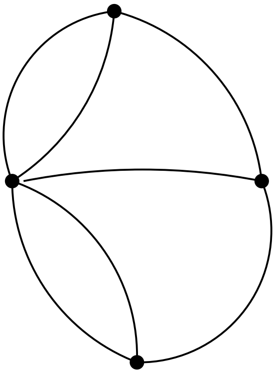
Actuel Kaliningrad, ville de Russie proche de la Lituanie et de la Pologne où coule la rivière Pregel



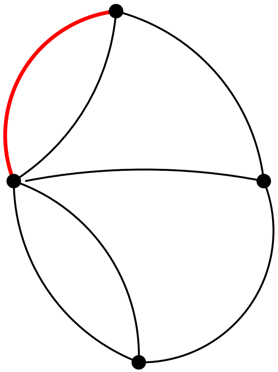


déterminer pour un multi-graphe donné (éventuellement orienté) s'il existe un circuit, i.e., un chemin fermé, passant une et une seule fois par chaque arête.

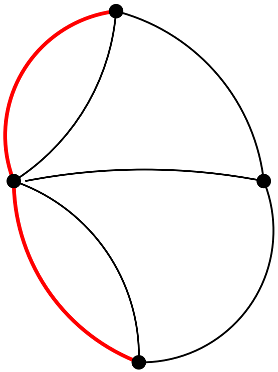
Partons du sommet 'en haut', est-ce que l'on s'y prend mal ?



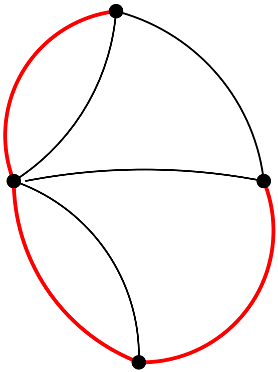
Partons du sommet 'en haut', est-ce que l'on s'y prend mal ?



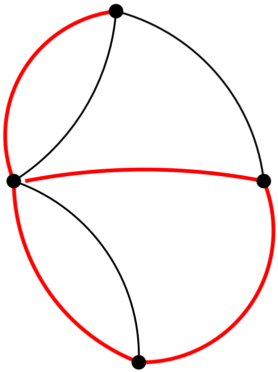
Partons du sommet 'en haut', est-ce que l'on s'y prend mal ?



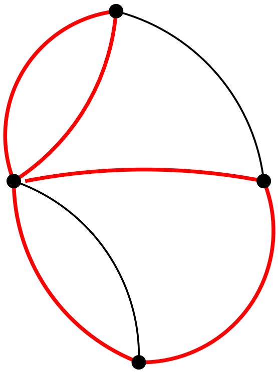
Partons du sommet 'en haut', est-ce que l'on s'y prend mal ?



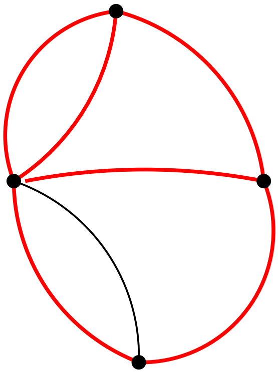
Partons du sommet 'en haut', est-ce que l'on s'y prend mal ?



Partons du sommet 'en haut', est-ce que l'on s'y prend mal ?

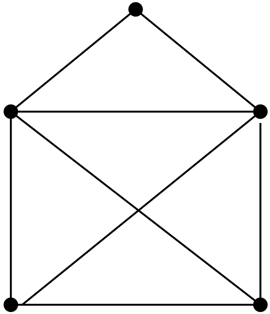


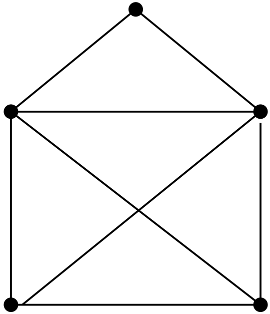
Partons du sommet 'en haut', est-ce que l'on s'y prend mal ?



Circuit eulérien VS chemin eulérien

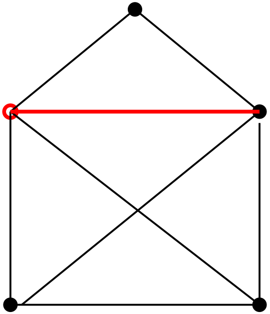
Chemin eulérien ? Tracer d'un trait...

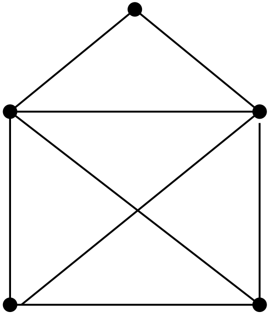




Circuit eulérien VS chemin eulérien

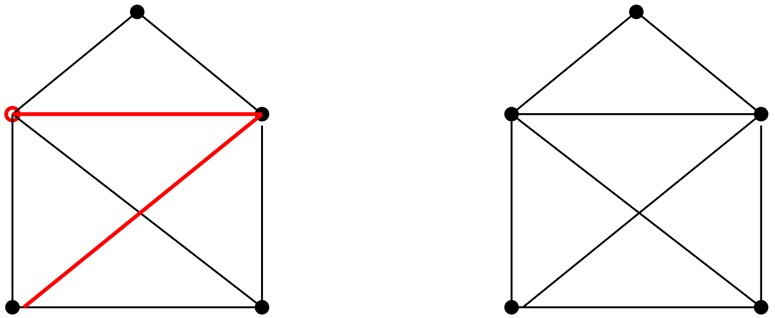
Chemin eulérien ? Tracer d'un trait. . .





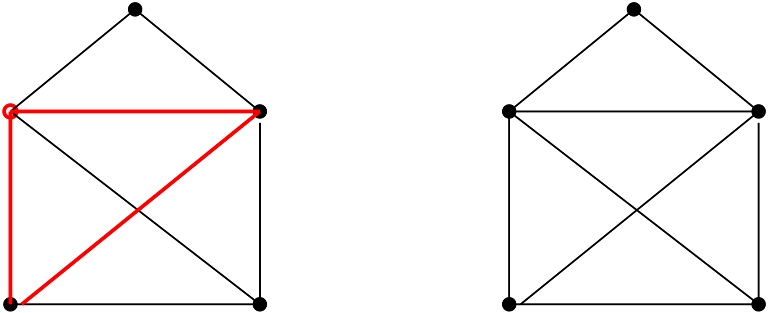
Circuit eulérien VS chemin eulérien

Chemin eulérien ? Tracer d'un trait. . .



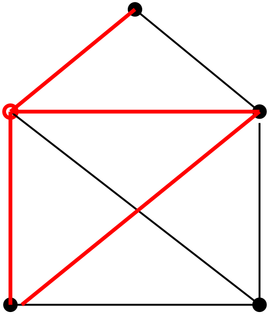
Circuit eulérien VS chemin eulérien

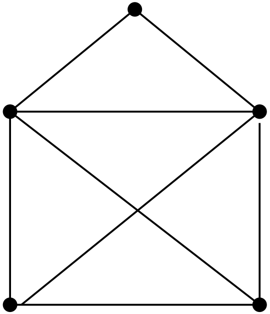
Chemin eulérien ? Tracer d'un trait...



Circuit eulérien VS chemin eulérien

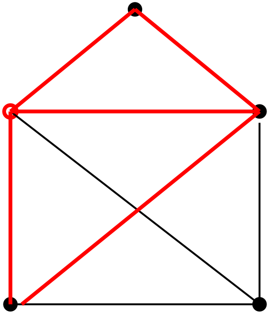
Chemin eulérien ? Tracer d'un trait...

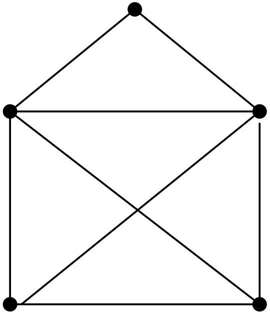




Circuit eulérien VS chemin eulérien

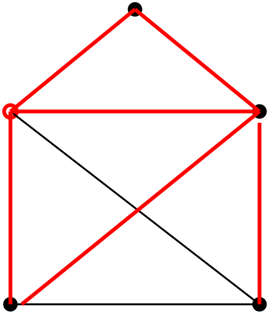
Chemin eulérien ? Tracer d'un trait...

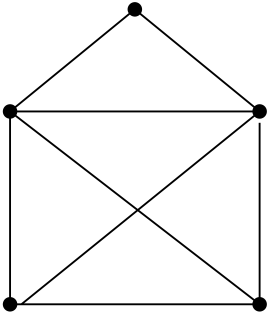




Circuit eulérien VS chemin eulérien

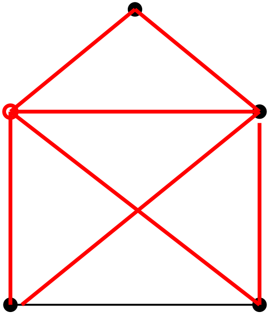
Chemin eulérien ? Tracer d'un trait. . .

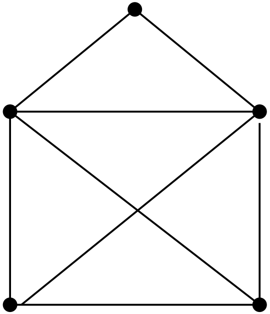




Circuit eulérien VS chemin eulérien

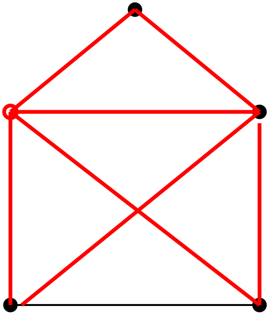
Chemin eulérien ? Tracer d'un trait. . .

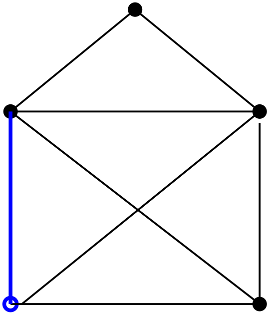




Circuit eulérien VS chemin eulérien

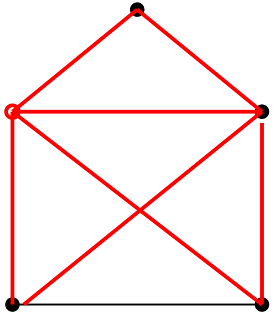
Chemin eulérien ? Tracer d'un trait. . .

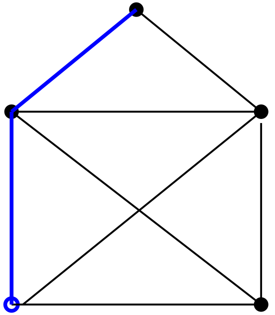




Circuit eulérien VS chemin eulérien

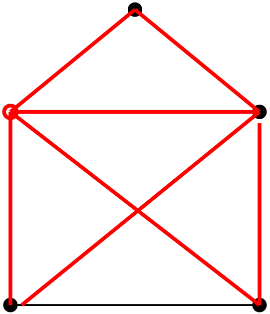
Chemin eulérien ? Tracer d'un trait...

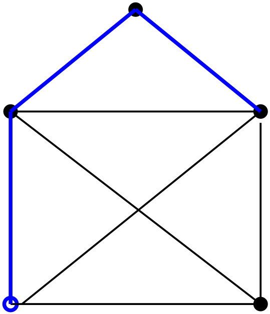




Circuit eulérien VS chemin eulérien

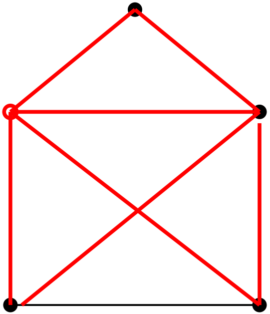
Chemin eulérien ? Tracer d'un trait. . .

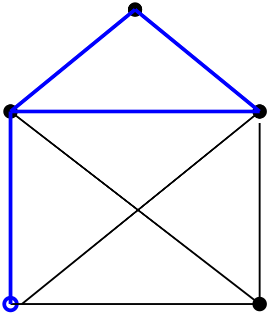




Circuit eulérien VS chemin eulérien

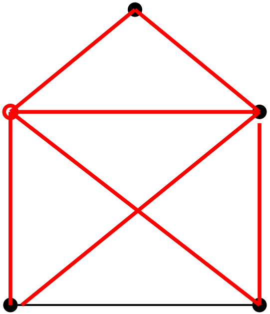
Chemin eulérien ? Tracer d'un trait. . .

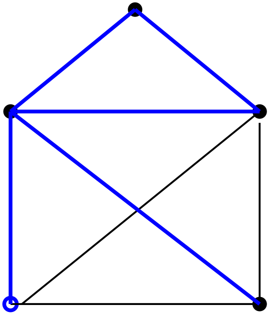




Circuit eulérien VS chemin eulérien

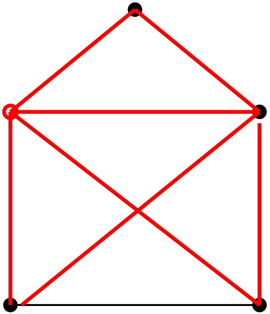
Chemin eulérien ? Tracer d'un trait...

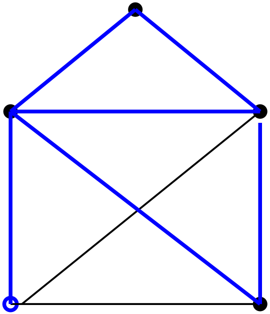




Circuit eulérien VS chemin eulérien

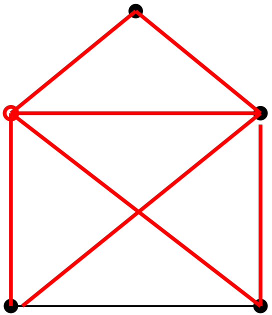
Chemin eulérien ? Tracer d'un trait...

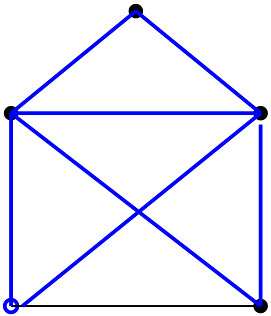




Circuit eulérien VS chemin eulérien

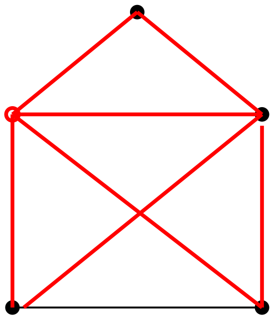
Chemin eulérien ? Tracer d'un trait...

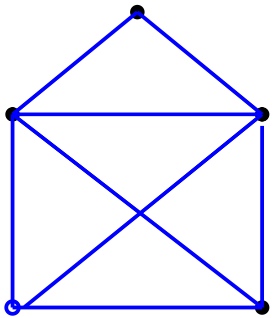




Circuit eulérien VS chemin eulérien

Chemin eulérien ? Tracer d'un trait...





Définition

Soit $G = (V, E)$ un multi-graphe non orienté.

Un chemin de longueur $k \geq 1$ est une suite ordonnée $(e_1, \ldots, e_k)$ de k arêtes adjacentes $e_i = \{e_{i,1}, e_{i,2}\}$,

$$\forall i \in \{1, \ldots, k-1\}, \quad e_{i,2} = e_{i+1,1}.$$

Ce chemin joint les sommets $e_{1,1}$ et $e_{k,2}$,
passe par les arêtes $e_1, \ldots, e_k$, les sommets $e_{1,1}, e_{1,2}, \ldots, e_{k,1}, e_{k,2}$.

Un chemin de longueur 0 joint toujours un sommet à lui-même.

Si $e_{1,1} = e_{k,2}$: cycle, circuit, chemin fermé

⚠ Terminologie variable suivant les auteurs et la langue...

Chemins

Définitions (suite)

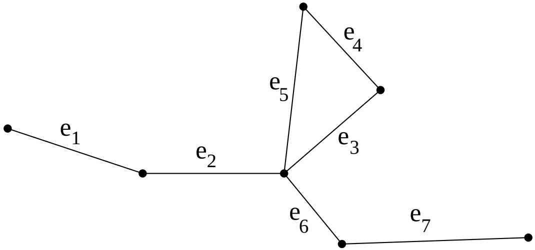
chemin dont les arêtes sont toutes distinctes :
piste ou chemin élémentaire

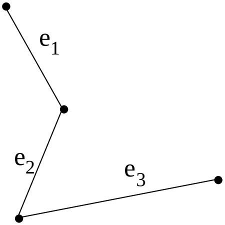
circuit dont les arêtes sont toutes distinctes :
piste fermée ou circuit élémentaire

chemin ne passant pas deux fois par un même sommet
(en particulier, toutes ses arêtes sont distinctes) : chemin simple

circuit ne passant pas deux fois par un même sommet — à l'exception du 'point de départ' : circuit simple

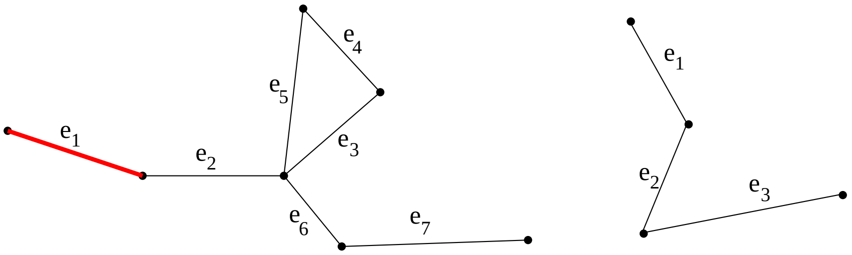
Piste (ou chemin élémentaire) — chemin simple





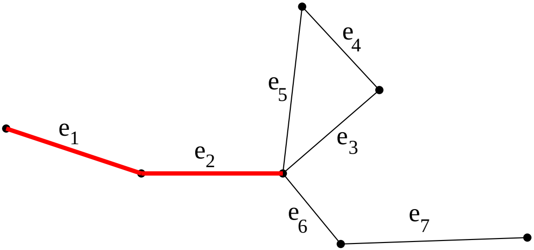
$(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ (e_1, e_2, e_3)

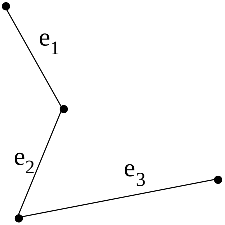
Piste (ou chemin élémentaire) — chemin simple



$(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ $\qquad$ (e_1, e_2, e_3)

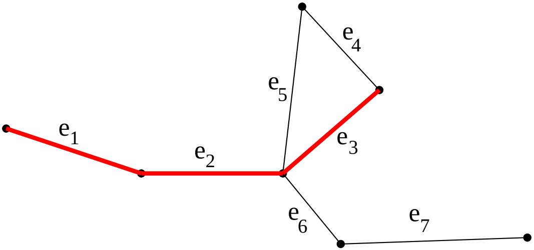
Piste (ou chemin élémentaire) — chemin simple

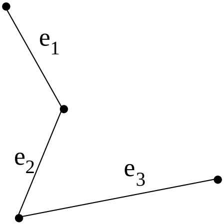




$(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ (e_1, e_2, e_3)

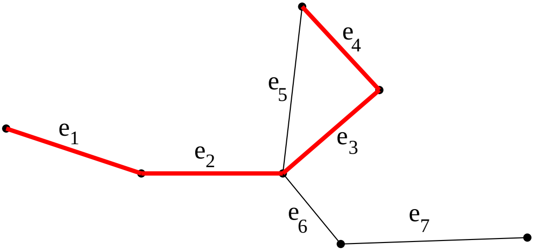
Piste (ou chemin élémentaire) — chemin simple

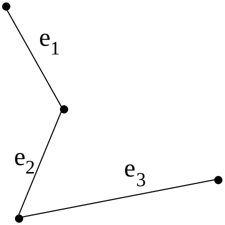




$(e_1, e_2, \textcolor{red}{e_3}, e_4, e_5, e_6, e_7)$ $\qquad$ (e_1, e_2, e_3)

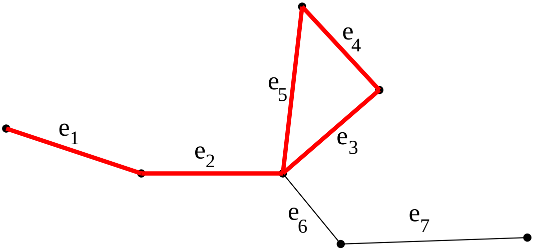
Piste (ou chemin élémentaire) — chemin simple

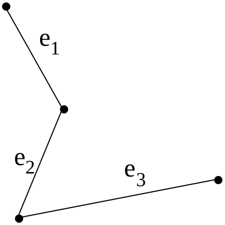




$(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ (e_1, e_2, e_3)

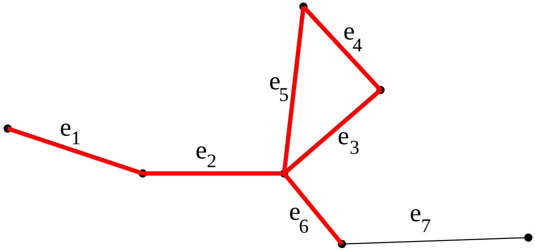
Piste (ou chemin élémentaire) — chemin simple

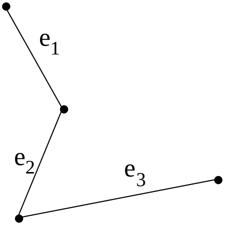




$(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ (e_1, e_2, e_3)

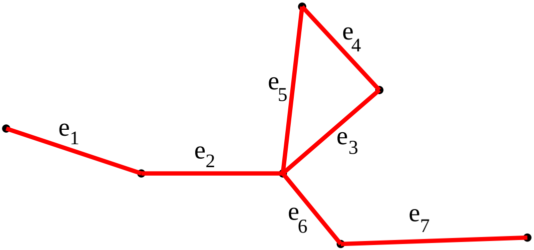
Piste (ou chemin élémentaire) — chemin simple

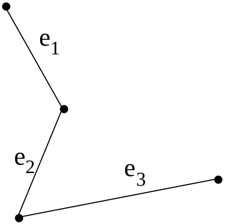




$(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ (e_1, e_2, e_3)

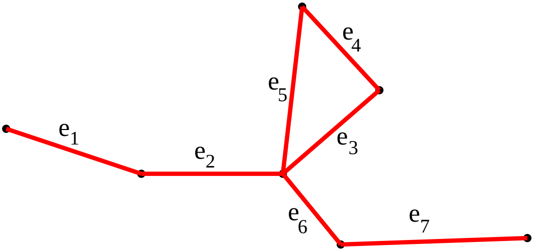
PISTE (OU CHEMIN ÉLÉMENTAIRE) — CHEMIN SIMPLE

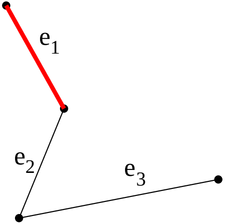




$(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ $\qquad$ (e_1, e_2, e_3)

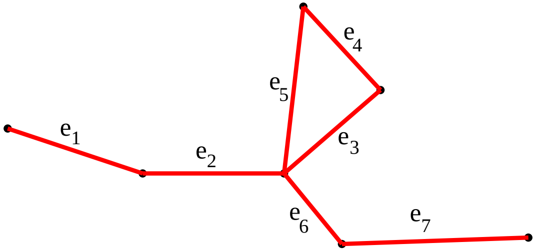
Piste (ou chemin élémentaire) — chemin simple

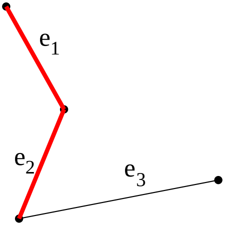




$(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ $\quad$ $({\color{red}e_1}, e_2, e_3)$

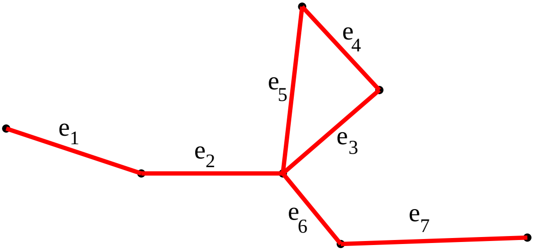
Piste (ou chemin élémentaire) — chemin simple

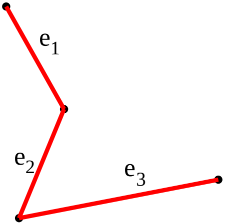




$(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ $\quad$ (e_1, e_2, e_3)

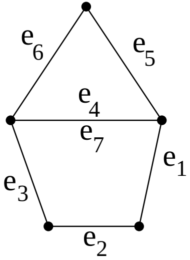
Piste (ou chemin élémentaire) — chemin simple

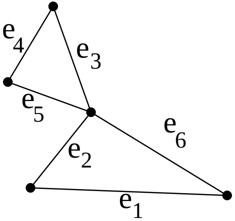


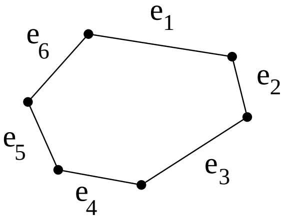


$(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ (e_1, e_2, e_3)

CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE

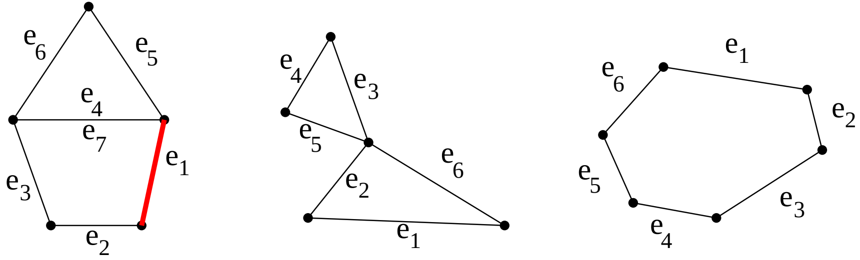






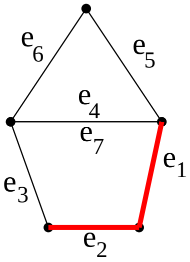
$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $(e_1, e_2, e_3, e_4, e_5, e_6)$

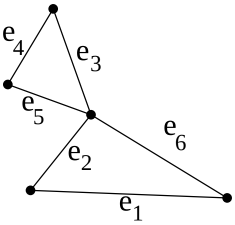
CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE

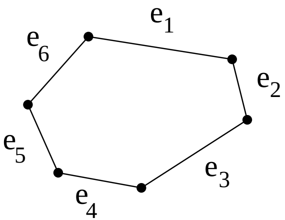


$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $\quad$ $(e_1, e_2, e_3, e_4, e_5, e_6)$

CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE

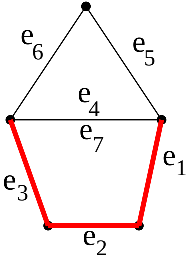


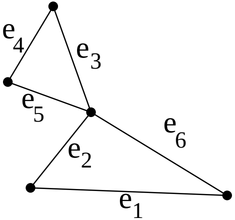


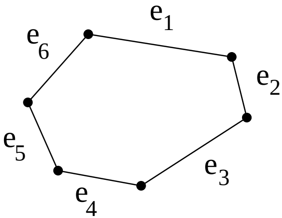


$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $\qquad$ $(e_1, e_2, e_3, e_4, e_5, e_6)$

CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE

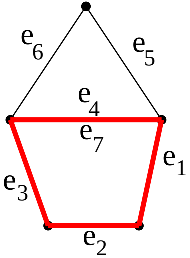


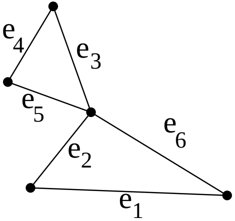


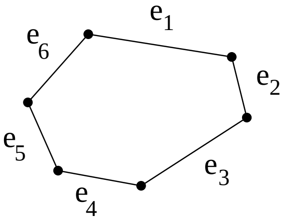


$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $\qquad$ $(e_1, e_2, e_3, e_4, e_5, e_6)$

CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE

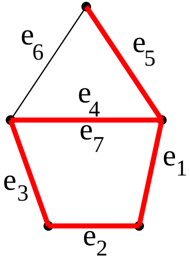


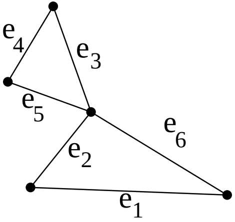


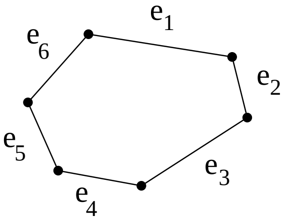


$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $\qquad$ $(e_1, e_2, e_3, e_4, e_5, e_6)$

CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE

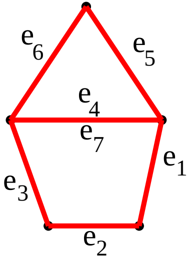


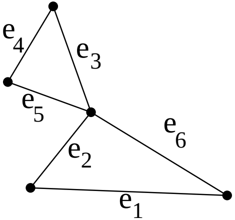


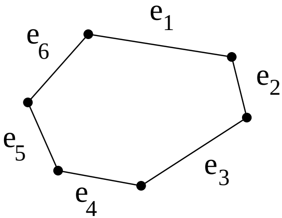


$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $\qquad$ $(e_1, e_2, e_3, e_4, e_5, e_6)$

CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE

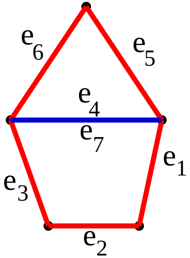


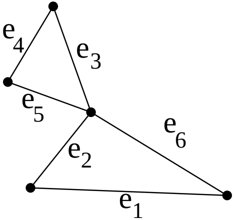


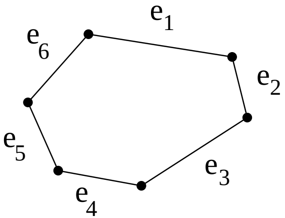


$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $\qquad$ $(e_1, e_2, e_3, e_4, e_5, e_6)$

CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE

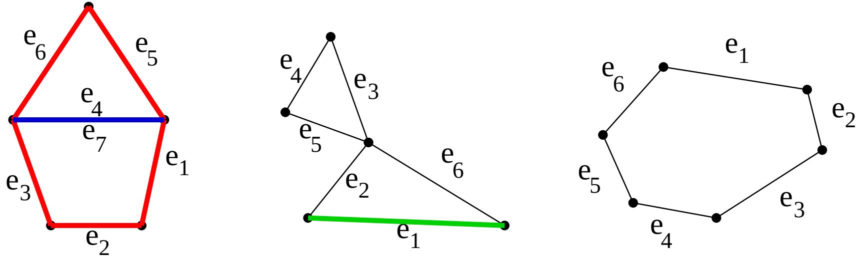






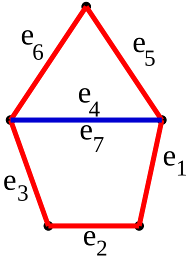
$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $\qquad$ $(e_1, e_2, e_3, e_4, e_5, e_6)$

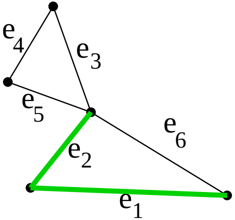
CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE

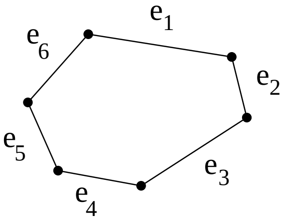


$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $(e_1, e_2, e_3, e_4, e_5, e_6)$

CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE

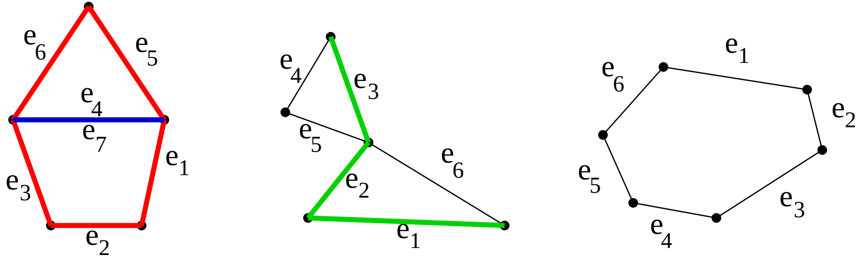






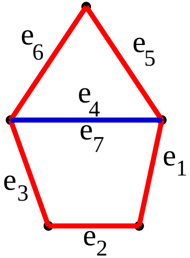
$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $\quad$ $(e_1, e_2, e_3, e_4, e_5, e_6)$

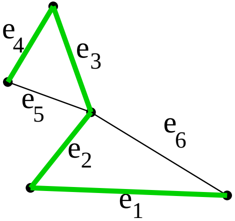
CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE

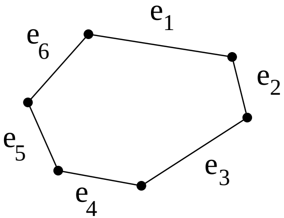


$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $\quad$ $(e_1, e_2, e_3, e_4, e_5, e_6)$

CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE

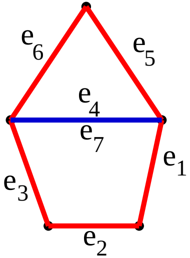


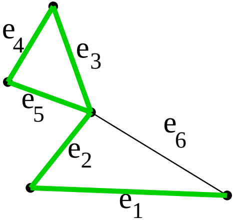


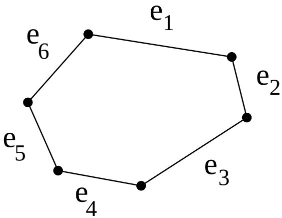


$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $\qquad$ $(e_1, e_2, e_3, {\color{red}e_4}, e_5, e_6)$

CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE

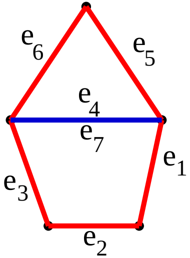


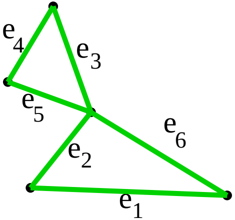


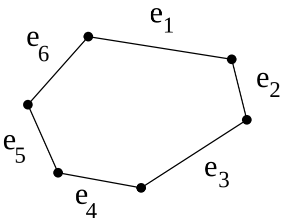


$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $\quad$ $(e_1, e_2, e_3, e_4, \color{red}{e_5}, e_6)$

CIRCUIT, PISTE FERMÉE ET CIRCUIT SIMPLE







$(e_1, e_2, e_3, e_4, e_5, e_6, e_4)$ $\quad$ $(e_1, e_2, e_3, e_4, e_5, e_6)$

Remarque

Dans le cas d'un graphe (qui n'est pas un multi-graphe), un chemin de longueur k est aussi univoquement déterminé par une *suite de sommets* $(v_0, \ldots, v_k)$ de manière telle que $\{v_i, v_{i+1}\}$ est une arête du graphe.

Connexité

Définition

Deux sommets a et b sont connectés,
s'il existe un chemin les joignant : $a \sim b$

Remarque : *un sommet est connecté à lui-même* (reflexivité)

composante connexe :

- classe d'équivalence pour $\sim$, ou,
- ensemble maximal (pour $\subseteq$) de sommets 2 à 2 connectés.

Un multi-graphe non orienté est connexe si

- $V/\sim$ contient une seule composante connexe, ou,
- $\forall a, b \in V$, $a \sim b$

Remarque : *on supposera que* $G = (\{v\}, \emptyset)$ *est connexe*

Chemins dans le cas orienté

Définition

⚠ $G = (V, E)$ multi-graphe <u>orienté</u>.
chemin de longueur $k \geq 1$: suite ordonnée

$$(e_1, \ldots, e_k)$$

de k arcs $e_i = (e_{i,1}, e_{i,2})$ t.q. $\forall i \in \{1, \ldots, k-1\}$, $e_{i,2} = e_{i+1,1}$.

Ce chemin joint les sommets $e_{1,1}$ et $e_{k,2}$.

S'il existe un chemin joignant deux sommets a et b : $a \rightarrow b$.

a et b sont fortement connectés ($a \leftrightarrow b$), si $a \rightarrow b$ et $b \rightarrow a$.

Remarque : *on impose* $a \leftrightarrow a$ *(réflexivité)*
"être fortement connecté" est une relation d'équivalence sur V*.*

Forte connexité

Définition

composante fortement connexe (ou f. connexe) de G :

- classe d'équivalence pour $\leftrightarrow$, ou,
- ensemble maximal M de sommets tels que pour $a, b \in M$, $a \rightarrow b$ (en particulier, $b \rightarrow a$)

$G = (V, E)$ est fortement connexe (ou f. connexe)

- $V/\leftrightarrow$ contient une seule classe, ou,
- $\forall a, b \in V$, $a \rightarrow b$ (en particulier, $b \rightarrow a$)

Remarque : les sommets appartenant à un cycle (élémentaire) maximal constituent une composante f. connexe.

Un multi-graphe orienté G est f. connexe SSI il existe un cycle passant par chaque sommet de celui-ci.

Simple connexité

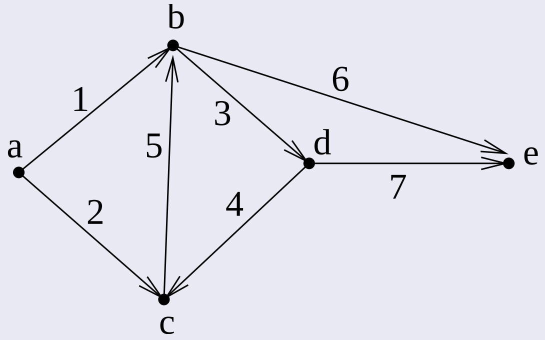
Définition

Si on supprime l'orientation des arcs de G et
si le multi-graphe non orienté obtenu est connexe,
alors G est simplement connexe (ou s. connexe).

$\leadsto$ composantes simplement connexes (ou s. connexes de G).

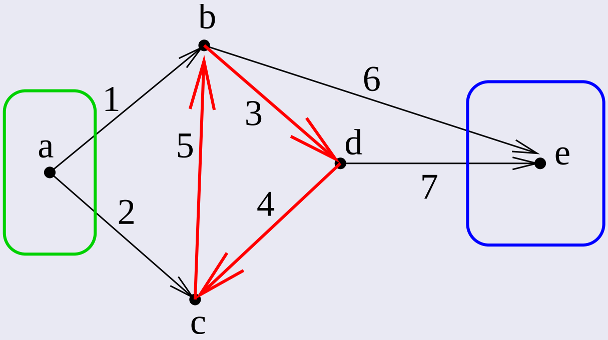
Connexité

Exemple, s. connexe ? f. connexe ?



Connexité

Exemple, s. connexe ? f. connexe ?

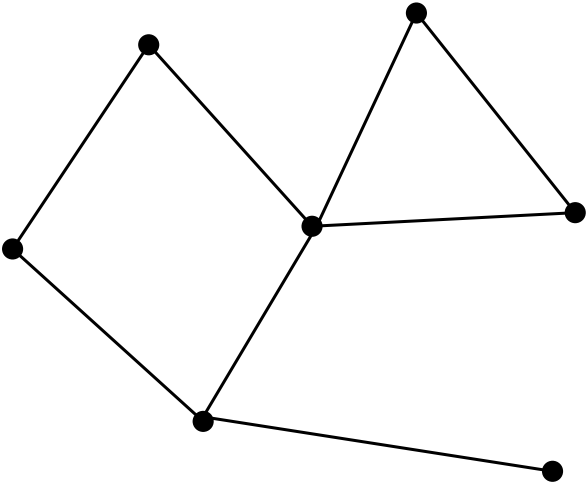


⚠ Test de connexité : un graphe simple suffit.

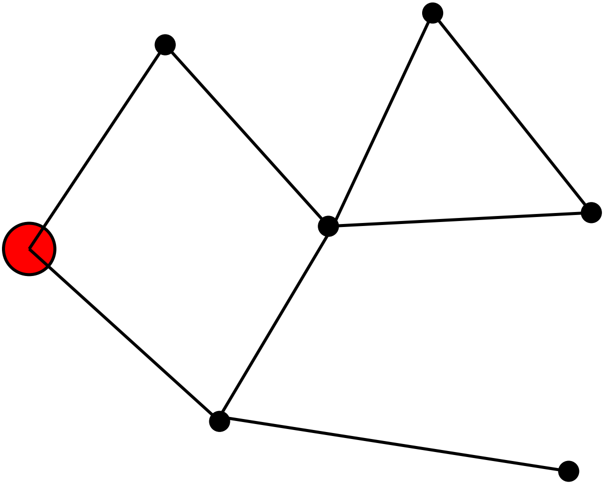
Algorithme naïf permettant de tester la connexité :

```
Choisir au hasard un sommet v0 ∈ V
Composante:= {v0},  New:= {v0}
Tant que New≠ ∅
   Voisins:= ∅
   pour tout sommet v appartenant à New
      Voisins:= Voisins ∪ ν(v)
   New:= Voisins\Composante
   Composante:= Composante ∪ New
Si Composante= V
   alors sortie : "oui, G connexe"
   sinon sortie : "non, G non connexe"
```

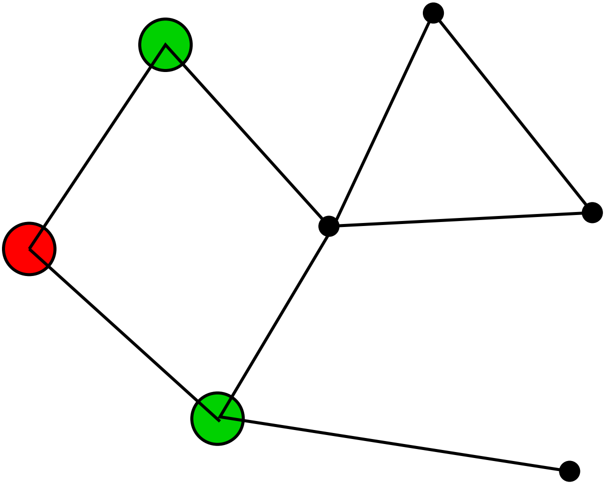
Connexité des graphes non orientés



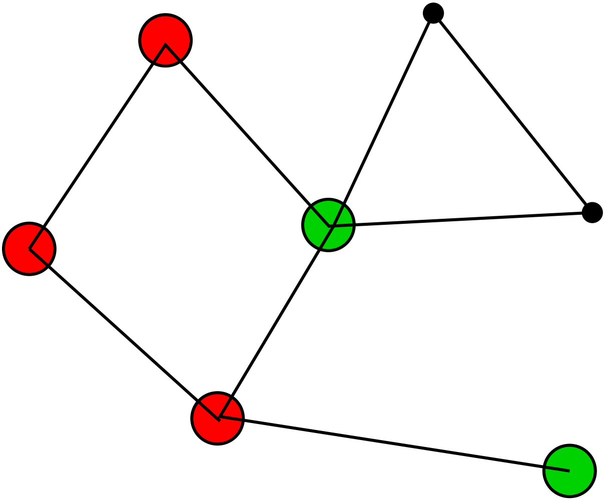
Connexité des graphes non orientés



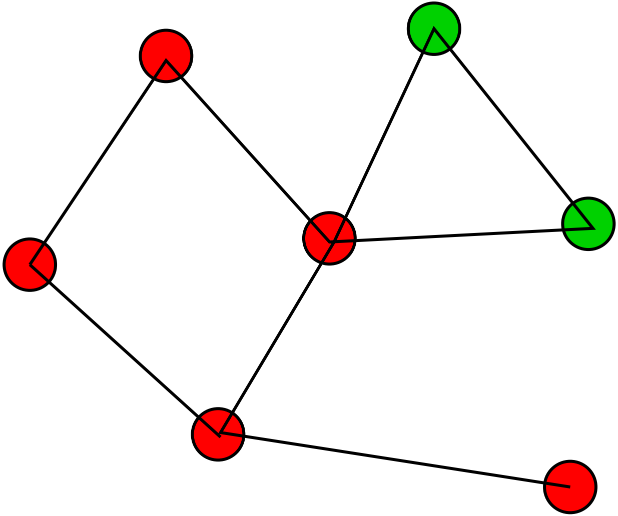
Connexité des graphes non orientés



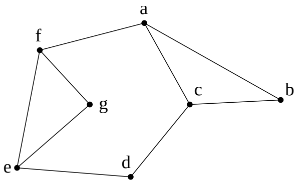
Connexité des graphes non orientés



Connexité des graphes non orientés



Connexité des graphes non orientés



représentation du graphe : dictionnaire des voisins / *adjacency list*

v	$\nu(v)$
a	$\{b, c, f\}$
b	$\{a, c\}$
c	$\{a, b, d\}$
d	$\{c, e\}$
e	$\{d, f, g\}$
f	$\{a, e, g\}$
g	$\{e, f\}$

Connexité des graphes non orientés

$v_0 = c$.

Composante	New	Voisins
$\{c\}$	$\{c\}$	$\emptyset$ $\nu(c) = \{a, b, d\}$
$\{c\} \cup \{a, b, d\}$ $= \{a, b, c, d\}$	$\{a, b, d\} \setminus \{c\}$ $= \{a, b, d\}$	$\emptyset$ $\nu(a) \cup \nu(b) \cup \nu(d)$ $= \{a, b, c, e, f\}$
$\{a, b, c, d\} \cup \{e, f\}$ $= \{a, b, c, d, e, f\}$	$\{a, b, c, e, f\} \setminus \{a, b, c, d\}$ $= \{e, f\}$	$\emptyset$ $\nu(e) \cup \nu(f)$ $= \{a, d, e, f, g\}$
$\{a, b, c, d, e, f\} \cup \{g\}$ $= \{a, b, c, d, e, f, g\}$	$\{a, d, e, f, g\} \setminus \{a, b, c, d, e, f\}$ $= \{g\}$	$\emptyset$ $\nu(g) = \{e, f\}$
$\{a, b, c, d, e, f, g\} \cup \{e, f\}$ $= \{a, b, c, d, e, f, g\}$	$\{e, f\} \setminus \{a, b, c, d, e, f, g\}$ $= \emptyset$	

Rappels :

- ensemble des arcs sortant de $v_i = \omega^+(v_i)$
- ensemble des successeurs de v : $\mathrm{succ}(v) = \{s_1, \dots, s_k\}$ sommets s_i tels que $(v, s_i) \in \omega^+(v)$, i.e., $(v, s_i) \in E$.
- ensemble des arcs entrant dans $v_j = \omega^-(v_j)$
- ensemble des prédécesseurs de v : $\mathrm{pred}(v) = \{s_1, \dots, s_k\}$ sommets s_i tels que $(s_i, v) \in \omega^-(v)$, i.e., $(s_i, v) \in E$.

Si W est un ensemble de sommets,

$$\mathrm{succ}(W) = \bigcup_{v \in W} \mathrm{succ}(v)$$

Clôture réflexive et transitive de succ

$$\mathrm{succ}^*(v) := \bigcup_{j=0}^{\infty} \mathrm{succ}^j(v) \quad \text{où} \quad \begin{cases} \mathrm{succ}^0(v) = v \\ \mathrm{succ}^{j+1}(v) = \mathrm{succ}(\mathrm{succ}^j(v)) \end{cases}.$$

$$\text{Si } G \text{ est fini,} \quad \mathrm{succ}^*(v) = \bigcup_{j=0}^{\#E} \mathrm{succ}^j(v).$$

S'il existe $k < \#E$ tel que $\mathrm{succ}^k(v) = \mathrm{succ}^{k+1}(v)$, alors

$$\mathrm{succ}^*(v) = \bigcup_{j=0}^{k} \mathrm{succ}^j(v).$$

$$\forall a, b \in V, \quad a \to b \Leftrightarrow b \in \mathrm{succ}^*(a).$$

Clôture réflexive et transitive de pred

Idem avec pred^*.

$$\forall a, b \in V, \quad b \to a \Leftrightarrow b \in \operatorname{pred}^*(a).$$

$$a \leftrightarrow b \Leftrightarrow b \in \operatorname{succ}^*(a) \cap \operatorname{pred}^*(a).$$

Décomposition en composantes f. connexes

L'algorithme précédent (test de connexité) s'adapte pour calculer $\mathrm{succ}^*(a)$ (resp. $\mathrm{pred}^*(a)$).

- Initialiser les variables `Composante` et `New` à $\{a\}$
- Remplacer $\nu(v)$ par $\mathrm{succ}(v)$ (resp. $\mathrm{pred}(v)$).

En recherchant l'intersection des deux ensembles,
on détermine alors la composante f. connexe du sommet a.

Remarque : *encore une fois, il suffit de regarder des graphes orientés simples.*

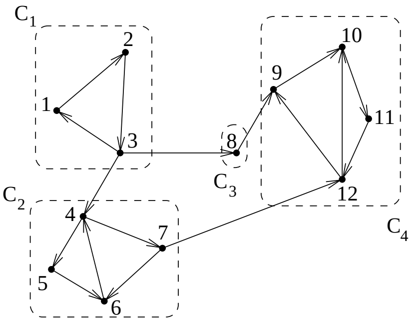
Décomposition en composantes f. connexes

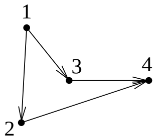
Définition

Soit $G = (V, E)$ un (multi-)graphe orienté.

Graphe *acyclique* des composantes ou condensé de G :

sommets $=$ les composantes f. connexes de G.
arc entre deux composantes f. connexes A et B,
s'il existe $a \in A$ et $b \in B$ tels que $a \to b$.





Décomposition en composantes f. connexes

Remarque

S'il existe $a \in A$ et $b \in B$ tels que $a \to b$,
alors il n'existe aucun $a' \in A$, $b' \in B$ tels que $b' \to a'$.

Sinon $A \cup B$ serait une composante f. connexe de G.
Absurde vu la maximalité des composantes connexes !

Conclusion : *le graphe des composantes est sans cycle*.

Graphes eulériens

Leonhard Euler (1707 – 1783)

Graphes eulériens

Définitions valides dans le cas orienté ou non,
graphe ou multigraphe :

Un circuit eulérien est un circuit passant une et une seule fois par chaque arête du graphe.

Un graphe eulérien est un graphe possédant un circuit eulérien.

Un chemin eulérien est un chemin passant une et une seule fois par chaque arête du graphe.

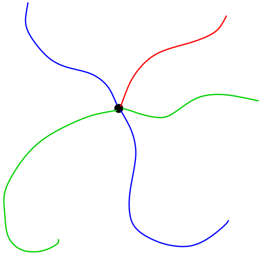
Graphes eulériens

⚠ On suppose avoir un graphe connexe.

Condition nécessaire

Pour que G soit eulérien, i.e. possède un circuit eulérien,
il est nécessaire que

- le degré de chaque sommet soit pair (cas non orienté)
- $d^-(v) = d^+(v)$ pour tout sommet v (cas orienté)



Graphes eulériens

Condition suffisante (propriété locale)

Théorème

Un multi-graphe fini non orienté connexe $G = (V, E)$ possède un circuit eulérien SSI le degré de chaque sommet est pair.

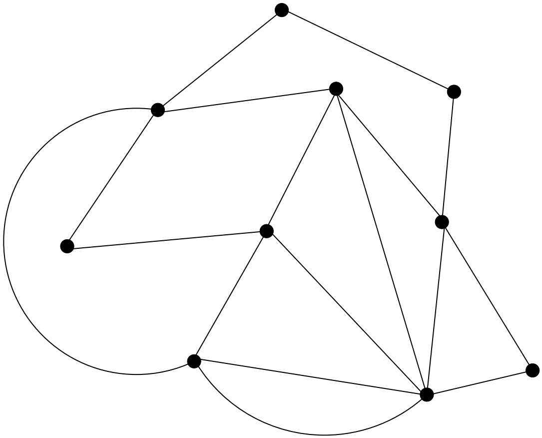
$\Rightarrow$ Hyp. : chaque sommet est de degré pair.

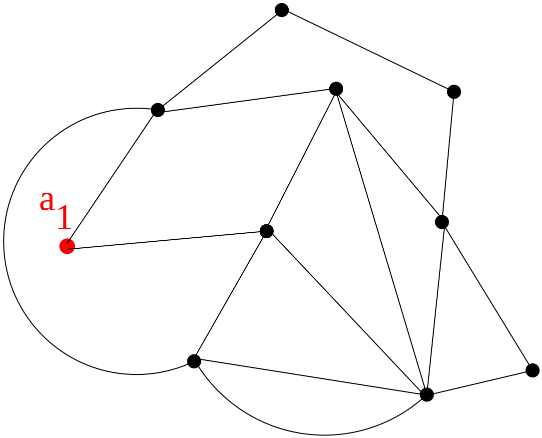
Construction d'une piste à partir d'un sommet a_1 de G.
A chaque étape $i \geq 1$, choix d'un sommet a_{i+1} t.q. une arête $\{a_i, a_{i+1}\} \in E$ est sélectionnée parmi les $\#E - i + 1$ arêtes non déjà sélectionnées.

• sélection toujours possible : chaque sommet est de degré pair, "*lorsqu'on aboutit dans un sommet, on peut toujours en repartir*".

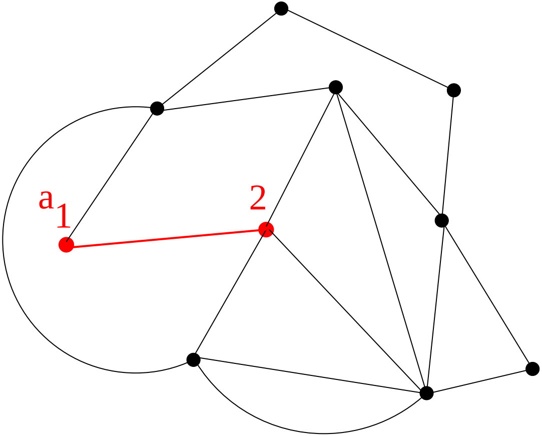
• la procédure s'achève : le graphe est fini.

Graphes eulériens

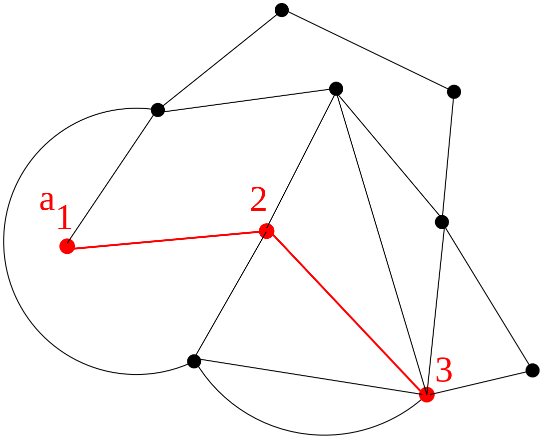



a_1

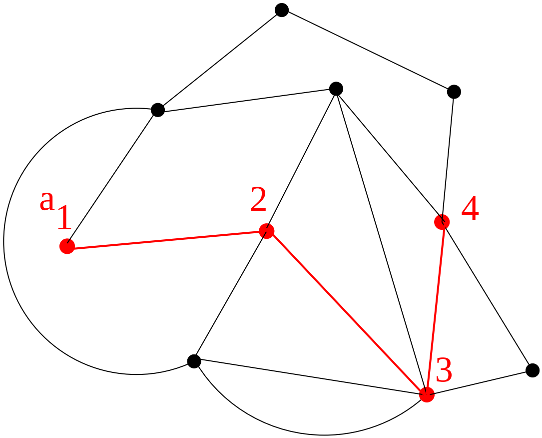
Graphes eulériens



Graphes eulériens



Graphes eulériens

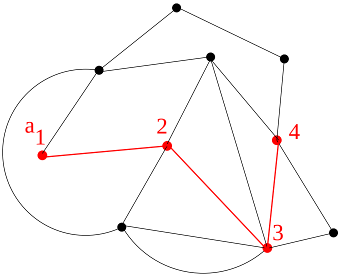


Graphes eulériens

On dispose d'une piste P joignant a_1 à un certain a_ℓ.

On peut supposer que cette piste est fermée, i.e., $a_\ell = a_1$.

Si a_ℓ diffère de a_1, puisque le degré de chaque sommet est pair, on peut étendre la piste en ajoutant une arête $\{a_\ell, a_{\ell+1}\}$. En continuant de la sorte, on épuise les sommets jusqu'à revenir en a_1.



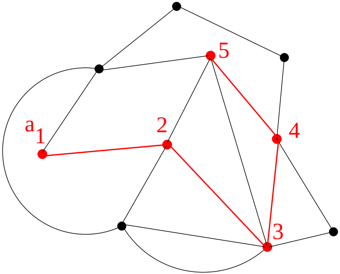
Si la piste fermée P est un circuit eulérien, le théorème est démontré.

Graphes eulériens

On dispose d'une piste P joignant a_1 à un certain a_ℓ.

On peut supposer que cette piste est fermée, i.e., $a_\ell = a_1$.

Si a_ℓ diffère de a_1, puisque le degré de chaque sommet est pair, on peut étendre la piste en ajoutant une arête $\{a_\ell, a_{\ell+1}\}$. En continuant de la sorte, on épuise les sommets jusqu'à revenir en a_1.

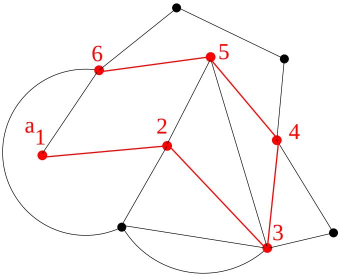


Si la piste fermée P est un circuit eulérien, le théorème est démontré.

On dispose d'une piste P joignant a_1 à un certain a_ℓ.

On peut supposer que cette piste est fermée, i.e., $a_\ell = a_1$.

Si a_ℓ diffère de a_1, puisque le degré de chaque sommet est pair, on peut étendre la piste en ajoutant une arête $\{a_\ell, a_{\ell+1}\}$. En continuant de la sorte, on épuise les sommets jusqu'à revenir en a_1.



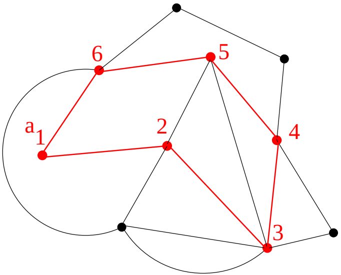
Si la piste fermée P est un circuit eulérien, le théorème est démontré.

Graphes eulériens

On dispose d'une piste P joignant a_1 à un certain a_ℓ.

On peut supposer que cette piste est fermée, i.e., $a_\ell = a_1$.

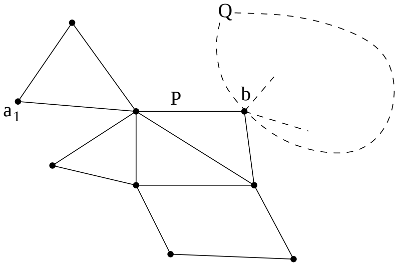
Si a_ℓ diffère de a_1, puisque le degré de chaque sommet est pair, on peut étendre la piste en ajoutant une arête $\{a_\ell, a_{\ell+1}\}$. En continuant de la sorte, on épuise les sommets jusqu'à revenir en a_1.



Si la piste fermée P est un circuit eulérien, le théorème est démontré.

GRAPHES EULÉRIENS

Sinon, il existe un sommet b de P qui est l'extrémité d'un nombre pair d'arêtes n'apparaissant pas dans P.

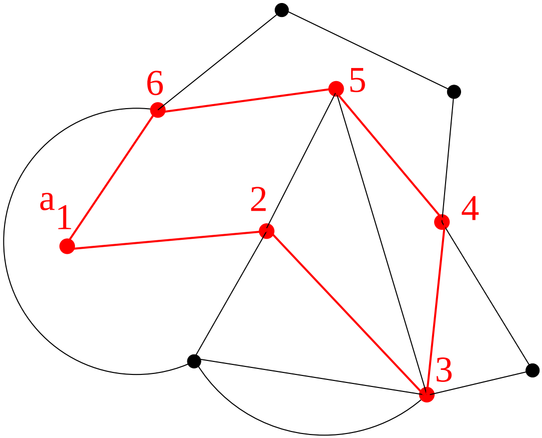


Depuis b, il est possible de construire une piste fermée Q formée uniquement d'arêtes n'apparaissant pas dans P. (même procédure, le degré de chaque sommet est encore pair.)

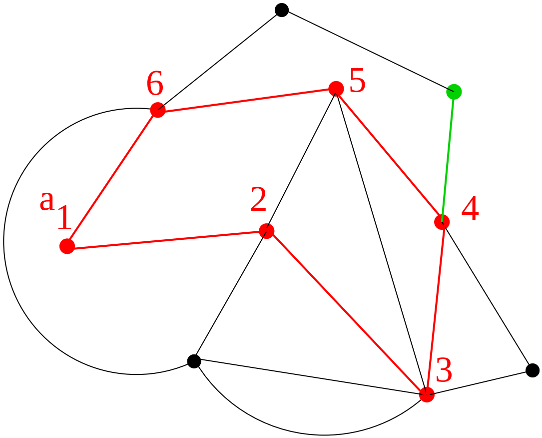
$\rightarrow$ on étend la piste P en une piste plus longue $P \cup Q$

$\rightarrow$ répéter cette étape un nombre fini de fois.

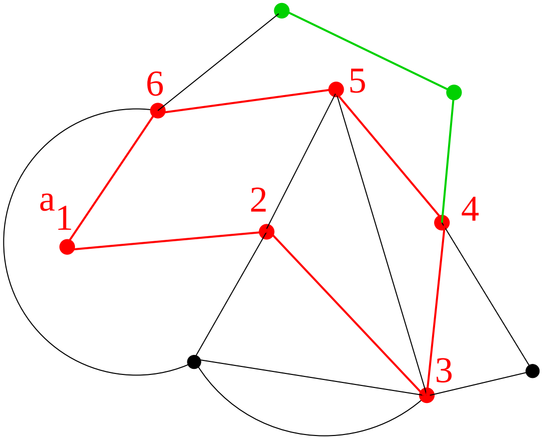
Graphes eulériens



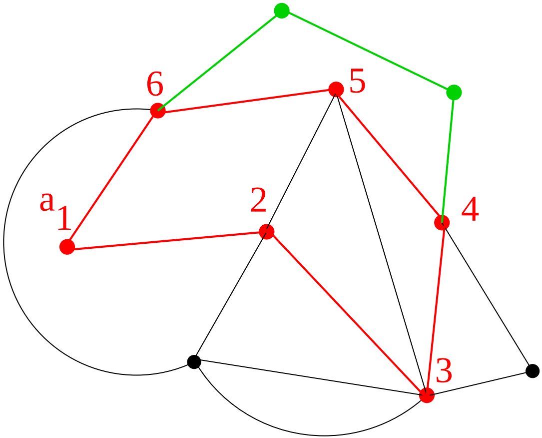
Graphes eulériens



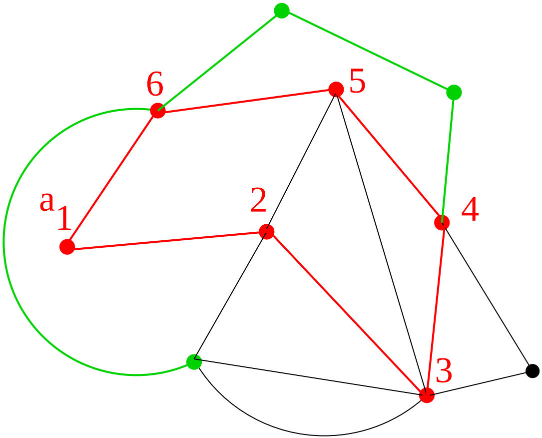
Graphes eulériens



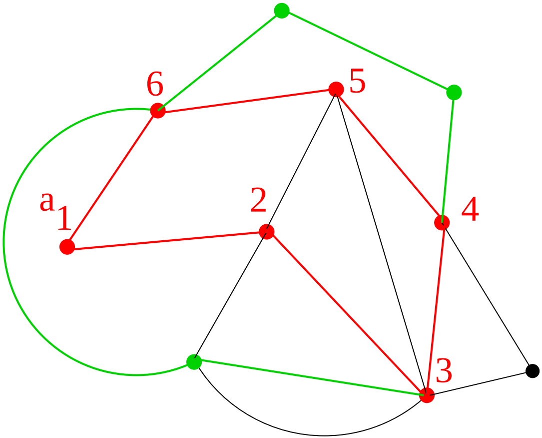
Graphes eulériens



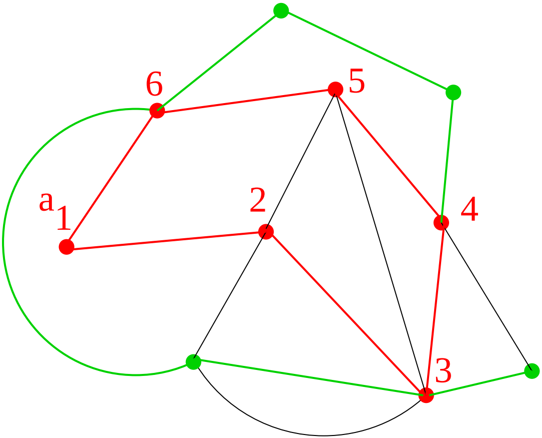
Graphes eulériens



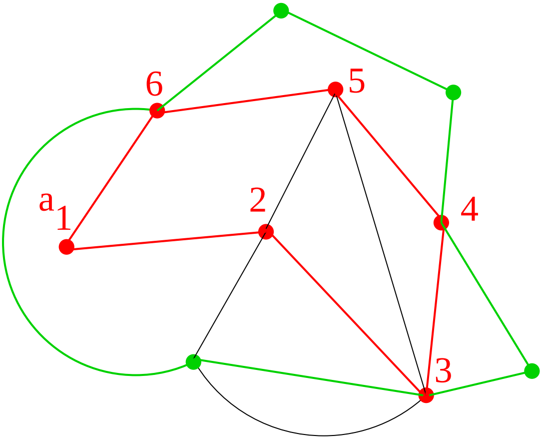
Graphes eulériens



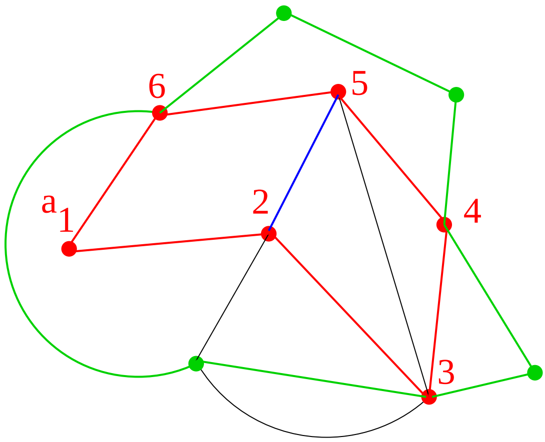
Graphes eulériens



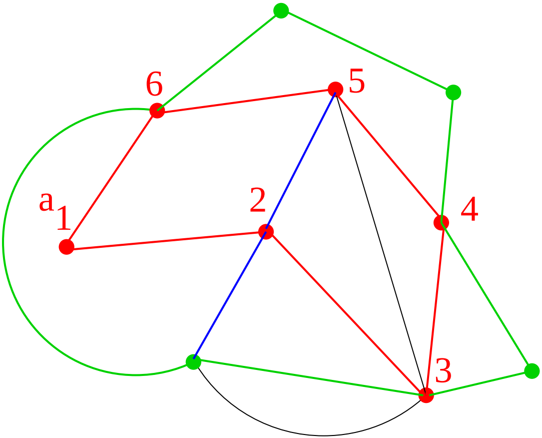
Graphes eulériens



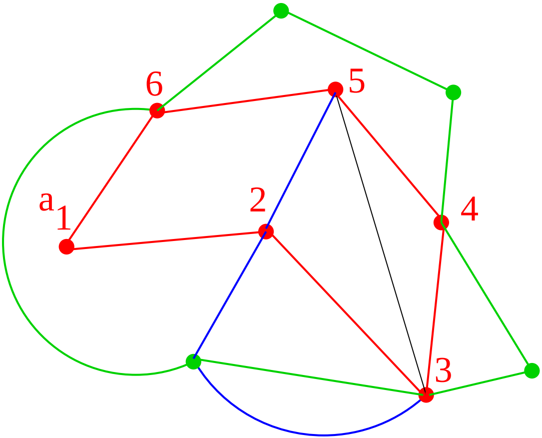
Graphes eulériens



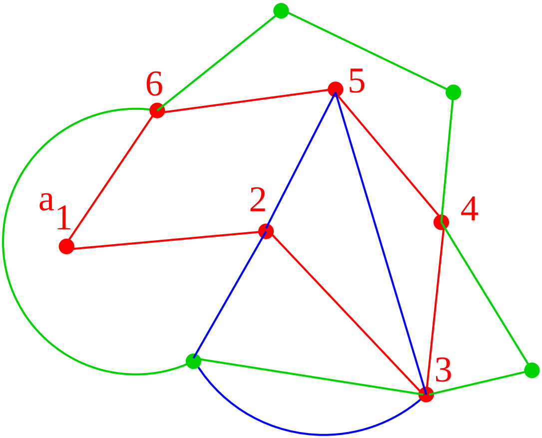
Graphes eulériens



Graphes eulériens



Graphes eulériens

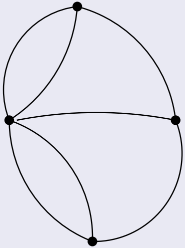


Remarque : tester si un graphe donné est eulérien, est un problème polynomial *par rapport à la taille de l'instance*

$\rightsquigarrow$ problème de décision $\in \mathcal{P}$

Corollaire

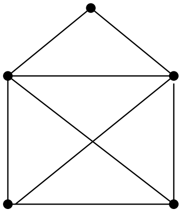
Le problème des sept ponts de Königsberg n'admet pas de solution.



Graphes eulériens

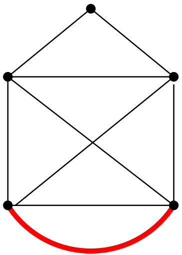
Corollaire

Un multi-graphe non orienté connexe possède un chemin eulérien joignant deux sommets a et b SSI a et b sont les deux seuls sommets de degré impair.



Corollaire

Un multi-graphe non orienté connexe possède un chemin eulérien joignant deux sommets a et b SSI a et b sont les deux seuls sommets de degré impair.



Graphes eulériens

Théorème

Un multi-graphe fini orienté s. connexe $G = (V, E)$ possède un circuit eulérien SSI $\forall v \in V$, $d^+(v) = d^-(v)$.

- Les graphes de De Bruijn sont eulériens.

Corollaire

Un multi-graphe fini orienté s. connexe $G = (V, E)$ possède un chemin eulérien SSI il existe deux sommets v_0 et v_1 tel que

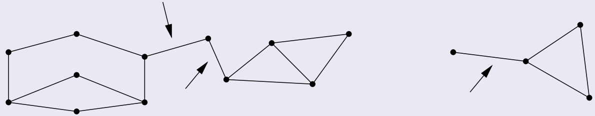
- pour tout $v \in V \setminus \{v_0, v_1\}$, $d^-(v) = d^+(v)$,
- $d^+(v_0) = d^-(v_0) + 1$,
- $d^-(v_1) = d^+(v_1) + 1$.

Algorithme alternatif pour rechercher un circuit eulérien

Définition

Soit $G = (V, E)$ un multi-graphe non orienté connexe
(ou une composante connexe d'un multi-graphe non orienté).

e est une arête de coupure si $G - e$ n'est plus connexe.



Algorithme de Fleury (1883) – peu efficace

```
Choisir un sommet v_0 ∈ V
i:=1
Répéter tant que possible
   Choisir une arête e_i = {v_{i-1}, v_i} ∈ V telle que
      ▸ e_i ≠ arêtes déjà choisies e_1, ..., e_{i-1}
      ▸ autant que possible, e_i ne doit pas être une
arête de coupure de G_i = G − {e_1, ..., e_{i-1}}
   i:=i + 1
```

Cet algorithme fournit une suite d'arêtes $e_1, e_2, \ldots$.

Corollaire

Soit G un graphe orienté (simple) t.q. $\forall v \in V$, $d^+(v) = d^-(v)$.
Alors, G est f. connexe SSI il est s. connexe.

f. connexe $\Rightarrow$ s. connexe.

$\Leftarrow$ G s. connexe et $\forall v \in V$, $d^+(v) = d^-(v)$
donc G possède un circuit eulérien,
on en conclut que G est f. connexe.

Distance
&
Algorithme de Dijkstra

Distance

Soit $G = (V, E)$ un multi-graphe non orienté connexe

(pour avoir une fonction totale, définie pour tout couple de sommets).

La distance entre a et $b \in V$, $\mathrm{d}(a, b)$, est
la longueur d'un plus court chemin joignant a et b.
Le diamètre de G est

$$\mathrm{diam}(G) = \max_{a,b \in V} \mathrm{d}(a, b).$$

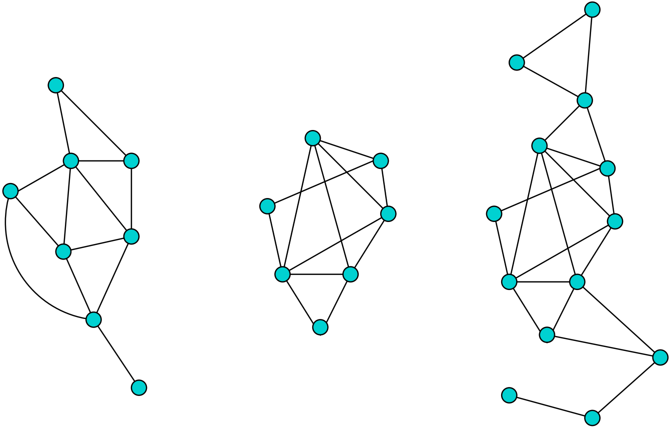
Si G est pondéré par $f : E \to \mathbb{R}^+$, la distance entre les sommets a et b est égale au poids minimal des chemins joignant a et b, i.e.,

$$\mathrm{d}(a, b) = \min_{\substack{\text{chemin } (e_1,\ldots,e_t) \\ \text{joignant } a \text{ et } b}} \sum_{i=1}^{t} f(e_i).$$

Rappel, en math. une distance vérifie

1. $d(\mathbf{x}, \mathbf{y}) \geq 0$,
2. $d(\mathbf{x}, \mathbf{y}) = 0$ si et seulement si $\mathbf{x} = \mathbf{y}$,
3. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$,
4. $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ (inégalité triangulaire),

Que vaut le diamètre des graphes suivants ?



Définition. . .

distance et diamètre s'adaptent aux multi-graphes orientés fortement connexes.

Cependant, la *fonction*

$$\mathrm{d}(\cdot,\cdot)$$

n'est en général pas symétrique.

Distance

Minimiser le nombre de “hops”

```
> traceroute www.google.be
traceroute to www.google.be (66.249.85.99), 30 hops max,
40 byte packets
1 mont3-0014.gw.ulg.ac.be (139.165.159.1)
2 segi3-0813-mont3.gw.ulg.ac.be (193.190.228.125)
3 inet3-3031.gw.ulg.ac.be (139.165.192.49)
4 fe.m20.access.liege.belnet.net (193.191.10.17)
5 oc48.m160.core.science.belnet.net (193.191.1.185)
6 oc192.m160.ext.science.belnet.net (193.191.1.2)
7 216.239.43.88
8 64.233.175.249
9 216.239.46.49
10 66.249.85.99
```

Recherche du plus court chemin

Soit $G = (V, E)$ un digraphe pondéré par $p : E \to \mathbb{R}^+$.
(L'algorithme s'applique aussi à un graphe non orienté.)

un plus court chemin = chemin de poids minimal

⚠ se restreindre à un graphe orienté simple

Remarque

On suppose p à valeurs dans $\mathbb{R}^+ \cup \{+\infty\}$,
on étend p de E à $V \times V$ en posant

- $p(x, x) = 0$, pour tout $x \in V$ et
- $p(x, y) = +\infty$, si $(x, y) \notin E$.

Algorithme de Dijkstra

Edsger W. Dijkstra (né en 1930)

Algorithme de Dijkstra

Intuitivement, v_1 est fixé, pour tout sommet v,

- $\mathtt{T}(v)$ initialisé à $p(v_1, v)$
- liste de sommets $\mathtt{C}(v)$ correspondant à un chemin de v_1 à v

Lorsque l'algorithme s'achève :

- $\mathtt{T}(v)$ contient le poids minimal des chemins joignant v_1 à v
- $\mathtt{C}(v)$ réalise un tel chemin

(ou alors, $\mathtt{T}(v) = +\infty$ si $v_1 \not\to v$).

Idée : faire grossir un ensemble $X \subseteq V$ t.q. un chemin de poids minimal de v_1 à $v \in X$ passe uniquement par des sommets de X.

X est initialisé à $\{v_1\}$ et à chaque étape,
on ajoute un sommet à l'ensemble, fin quand $X = V$.

Algorithme de Dijkstra

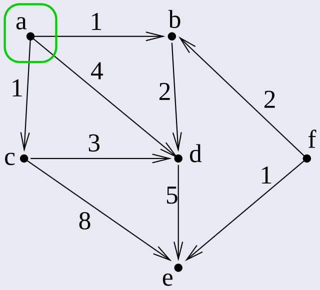
Pour tout $v \in V$, T(v):=$p(v_1, v)$, C(v):=(v_1, v)
X:=$\{v_1\}$
Tant que X$\neq V$, répéter
  Choisir $v \in V\backslash$X t.q. $\forall y \in V\backslash$X, T$(v) \leq$T$(y)$
  X:=X$\cup\{v\}$
  Pour tout $y \in V\backslash$X
    Si T(y)>T$(v) + p(v, y)$,
      alors T(y):=T$(v) + p(v, y)$ et C(y):=[C(v),y]

Algorithme de Dijkstra

Dijkstra(G, p, v_1) avec $G = (V, E)$ graphe simple, p fonction de poids et $v_1 \in V$

for all $v \in V \setminus \{v_1\}$,
    do $T(v) \leftarrow \mathsf{p}(v_1, v)$;
        if $T(v) \neq +\infty$
            then $C(v) \leftarrow (v_1, v)$
            else $C(v) \leftarrow ()$;
$X \leftarrow \{v_1\}$;
while $X \neq V$
    do pick $v \in V \setminus X$ s.t. $\forall y \in V \setminus X,\ T(v) \leq T(y)$;
        $X \leftarrow X \cup \{v\}$;
        for all $y \in V \setminus X$,
            do if $T(y) > T(v) + \mathsf{p}(v, y)$
                then $T(y) \leftarrow T(v) + \mathsf{p}(v, y)$;
                    $C(y) \leftarrow \mathtt{concat}(C(v), y)$;

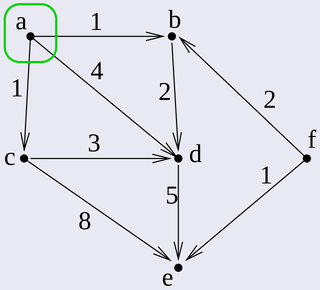
Exemple



$X = \{a\}$

$$\begin{array}{c|cccccc}
v & \color{blue}{a} & b & c & d & e & f \\
\hline
\mathtt{T}(v) & 0 & 1 & 1 & 4 & +\infty & +\infty \\
\mathtt{C}(v) & (a,a) & (a,b) & (a,c) & (a,d) & (a,e) & (a,f)
\end{array}.$$

Exemple

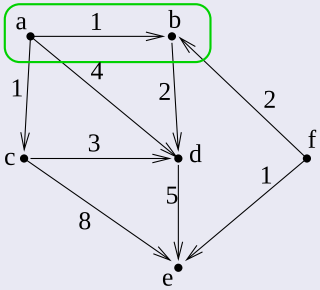


$\mathtt{X} = \{a\}$

v	a	b	c	d	e	f
$\mathtt{T}(v)$	0	1	1	4	$+\infty$	$+\infty$
$\mathtt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, d)	(a, e)	(a, f)

.

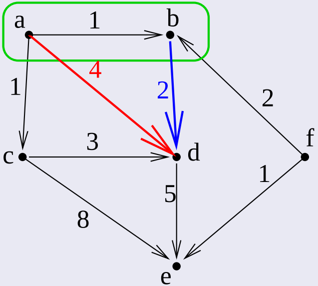
Exemple



$\mathtt{X} = \{a, b\}$

$$\begin{array}{c|cccccc} v & a & b & c & d & e & f \\ \hline \mathtt{T}(v) & 0 & 1 & 1 & 4 & +\infty & +\infty \\ \mathtt{C}(v) & (a,a) & (a,b) & (a,c) & (a,d) & (a,e) & (a,f) \end{array}.$$

Exemple



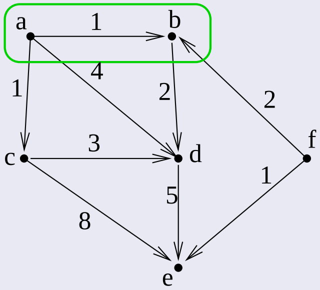
$\mathtt{X} = \{a, b\}$

v	a	b	c	d	e	f
$\mathtt{T}(v)$	0	1	1	4	$+\infty$	$+\infty$
$\mathtt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, d)	(a, e)	(a, f)

.

Algorithme de Dijkstra

Exemple



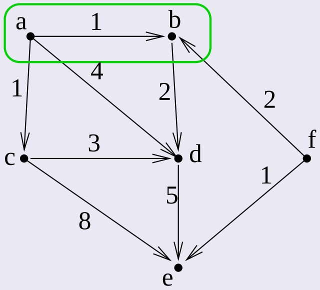
$\mathtt{X} = \{a, b\}$

v	$\color{blue}{a}$	$\color{blue}{b}$	c	d	e	f
$\mathtt{T}(v)$	0	1	1	3	$+\infty$	$+\infty$
$\mathtt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, e)	(a, f)

.

Algorithme de Dijkstra

Exemple

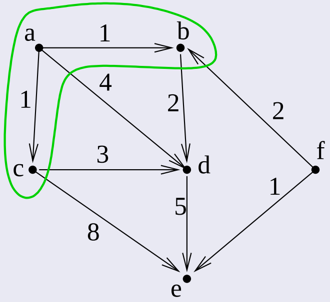


$\mathtt{X} = \{a, b\}$

v	a	b	c	d	e	f
$\mathtt{T}(v)$	0	1	1	3	$+\infty$	$+\infty$
$\mathtt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, e)	(a, f)

.

Exemple



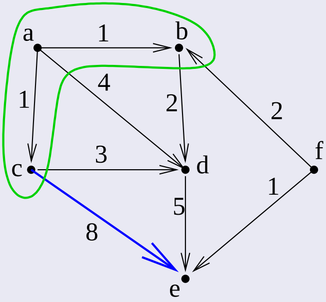
$\texttt{X} = \{a, b, c\}$

v	a	b	c	d	e	f
$\texttt{T}(v)$	0	1	1	3	$+\infty$	$+\infty$
$\texttt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, e)	(a, f)

.

Algorithme de Dijkstra

Exemple



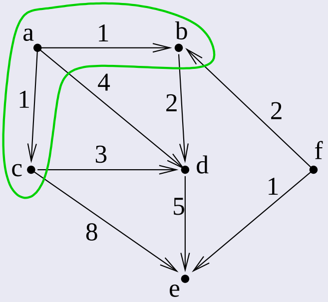
$\mathtt{X} = \{a, b, c\}$

v	a	b	c	d	e	f
$\mathtt{T}(v)$	0	1	1	3	$+\infty$	$+\infty$
$\mathtt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, e)	(a, f)

.

Algorithme de Dijkstra

Exemple

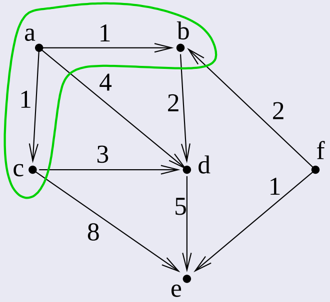


$\mathtt{X} = \{a, b, c\}$

v	a	b	c	d	e	f
$\mathtt{T}(v)$	0	1	1	3	9	$+\infty$
$\mathtt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, c, e)	(a, f)

.

Exemple

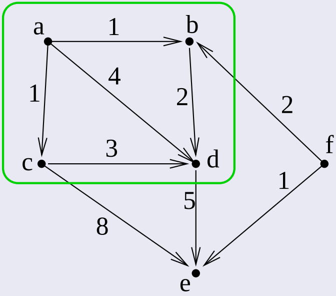


$\mathtt{X} = \{a, b, c\}$

v	a	b	c	d	e	f
$\mathtt{T}(v)$	0	1	1	3	9	$+\infty$
$\mathtt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, c, e)	(a, f)

.

Exemple



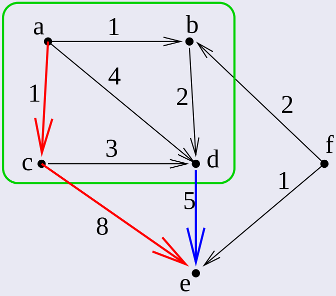
$\mathtt{X} = \{a, b, c, d\}$

v	a	b	c	d	e	f
$\mathtt{T}(v)$	0	1	1	3	9	$+\infty$
$\mathtt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, c, e)	(a, f)

.

Algorithme de Dijkstra

Exemple

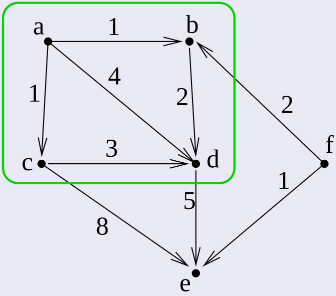


$\mathtt{X} = \{a, b, c, d\}$

v	a	b	c	d	e	f
$\mathtt{T}(v)$	0	1	1	3	9	$+\infty$
$\mathtt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, c, e)	(a, f)

.

Exemple



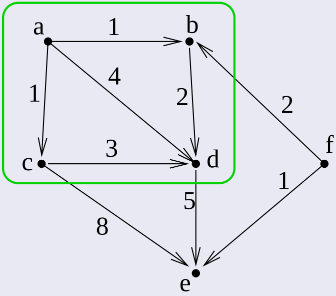
$\mathtt{X} = \{a, b, c, d\}$

v	a	b	c	d	e	f
$\mathtt{T}(v)$	0	1	1	3	8	$+\infty$
$\mathtt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, b, d, e)	(a, f)

.

Algorithme de Dijkstra

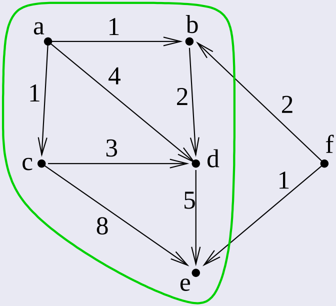
Exemple



$\mathtt{X} = \{a, b, c, d\}$

$$
\begin{array}{c|cccccc}
v & \color{blue}{a} & \color{blue}{b} & \color{blue}{c} & \color{blue}{d} & e & f \\
\hline
\mathtt{T}(v) & 0 & 1 & 1 & 3 & \color{red}{8} & +\infty \\
\mathtt{C}(v) & (a, a) & (a, b) & (a, c) & (a, b, d) & (a, b, d, e) & (a, f)
\end{array}.
$$

Exemple



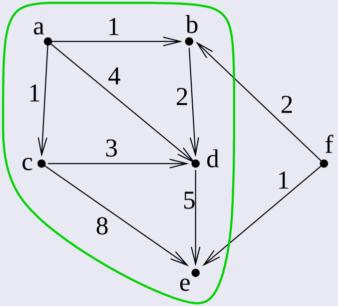
$\mathtt{X} = \{a, b, c, d, e\}$

v	a	b	c	d	e	f
$\mathtt{T}(v)$	0	1	1	3	8	$+\infty$
$\mathtt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, b, d, e)	(a, f)

.

Algorithme de Dijkstra

Exemple



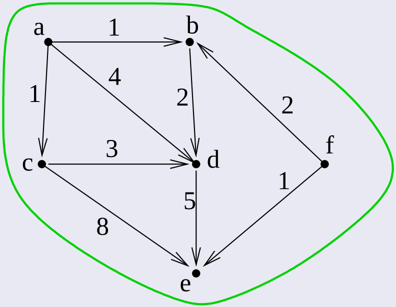
$\mathtt{X} = \{a, b, c, d, e\}$

v	a	b	c	d	e	f
$\mathtt{T}(v)$	0	1	1	3	8	$+\infty$
$\mathtt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, b, d, e)	(a, f)

.

Algorithme de Dijkstra

Exemple



$\mathtt{X} = \{a, b, c, d, e, f\}$

v	a	b	c	d	e	f
$\mathtt{T}(v)$	0	1	1	3	8	$+\infty$
$\mathtt{C}(v)$	(a, a)	(a, b)	(a, c)	(a, b, d)	(a, b, d, e)	(a, f)

.

Algorithme de Dijkstra

Dijkstra(G, p, v_1) avec $G = (V, E)$ graphe simple, p fonction de poids et $v_1 \in V$

for all $v \in V \setminus \{v_1\}$,
    do $T(v) \leftarrow \mathsf{p}(v_1, v)$;
        if $T(v) \neq +\infty$
            then $C(v) \leftarrow (v_1, v)$
            else $C(v) \leftarrow ()$;
$X \leftarrow \{v_1\}$;
while $X \neq V$
    do pick $v \in V \setminus X$ s.t. $\forall y \in V \setminus X,\ T(v) \leq T(y)$;
        $X \leftarrow X \cup \{v\}$;
        for all $y \in V \setminus X$,
            do if $T(y) > T(v) + \mathsf{p}(v, y)$
                then $T(y) \leftarrow T(v) + \mathsf{p}(v, y)$;
                    $C(y) \leftarrow \mathtt{concat}(C(v), y)$;

Preuve de l'algorithme

1. l'algorithme s'achève,
2. s'il s'achève, il s'achève avec le bon résultat.

X_n = ensemble X à la $n$ième itération
$\rightsquigarrow X_1 = \{v_1\}$ et $\#X_n = n$

v_{n+1} sommet choisi à la ligne 8 : $X_{n+1} \setminus X_n = \{v_{n+1}\}$

$T_n(y)$ = valeur de la variable $T(y)$ à la $n$ième itération
Au vu des lignes 11–12, on a toujours

$$T_{n+1}(y) \leq T_n(y)$$

Preuve de l'algorithme

Récurrence sur n, on va montrer que

i) $\forall v \in X_n,\ T_n(v)$ est le poids minimal de *tous* les chemins joignant v_1 à v.

ii) $\forall v \notin X_n,\ T_n(v)$ est le poids minimal des chemins joignant v_1 à v qui, à l'exception de v, passent uniquement par des sommets de X_n.

D'où le résultat pour $n = \# V$.

Cas de base, $n = 1$, OK vu les lignes 1–5

Supposons que i) et ii) sont vérifiés pour $1 \leq n < \# V$ et vérifions-le pour $X_{n+1} = X_n \cup \{v_{n+1}\}$.

Preuve de l'algorithme

Supposons que i) n'est pas vérifié pour X_{n+1}.
Mais i) est vérifié pour X_n (hyp. de récurrence).

Donc, i) n'est pas vérifié pour le sommet v_{n+1} :
$\rightsquigarrow$ *il existe un chemin C de v_1 à v_{n+1} de poids $p < T_{n+1}(v_{n+1})$.*

Mais ii) est vérifié pour X_n et $v_{n+1} \notin X_n$:
$\rightsquigarrow$ *$T_n(v_{n+1})$ est le poids minimal des chemins joignant v_1 à v_{n+1} qui, à l'exception de v_{n+1}, passent uniquement par des sommets de X_n.*
De plus,

$$p < T_{n+1}(v_{n+1}) \leq T_n(v_{n+1})$$

Conclusion : C doit passer par un sommet $\notin X_{n+1}$.
Soit u le 'premier' tel sommet de C. Vu ii), il vérifie $T_n(u) \leq p$.

Donc $T_n(u) < T_n(v_{n+1})$ contredit la ligne 8 !

Preuve de l'algorithme

ii) est vérifié pour X_n (hyp. de récurrence).
$\rightsquigarrow$ si $u \notin X_n$, $T_n(u)$ est le poids minimal des chemins joignant v_1 à u qui, à l'exception de u, passent uniquement par des sommets de X_n.

On passe de X_n à X_{n+1}, que contient $T_{n+1}(u)$ pour $u \notin X_{n+1}$?

On regarde les chemins joignant v_1 à u qui, à l'exception de u, passent uniquement par des sommets de X_{n+1} :

- ceux ne passant par v_{n+1}
- ceux passant par v_{n+1}, vu i) il suffit de considérer ceux où v_{n+1} apparaît uniquement comme 'dernier' sommet

La conclusion suit des lignes 10–12

Algorithme de Dijkstra

Complexité générique :

$$\mathcal{O}(\#E + (\#V)^2) = \mathcal{O}((\#V)^2)$$

- implémentation efficace $\mathcal{O}(\#E + \#V . \log(\#V))$
- graphes particuliers (peu denses)
- complexité moyenne
- encore d'autres variantes...

Pointeurs : A* search algorithm, Bellman–Ford algorithm, Johnson's algorithm, Floyd–Warshall algorithm

Algorithme de Dijkstra

Routage de paquets sur Internet. Lorsque chaque paquet passe par une passerelle, un routeur étudie où le paquet doit se rendre et décide alors de la passerelle suivante où l'envoyer. Le but de chaque routeur est de propager un paquet vers un point de plus en plus près de sa destination finale.

Un des types de routage : routage SPF (Shortest Path First) chaque routeur gère sa propre carte de l'internet afin de mettre à jour sa table de routage en calculant les plus courts chemins entre lui et les autres destinations. Sa carte est un graphe orienté et pondéré, dont les sommets sont les passerelles et les arcs les connexions entre celles-ci. Chaque arc est pondéré par les dernières performances constatées sur la connexion.