

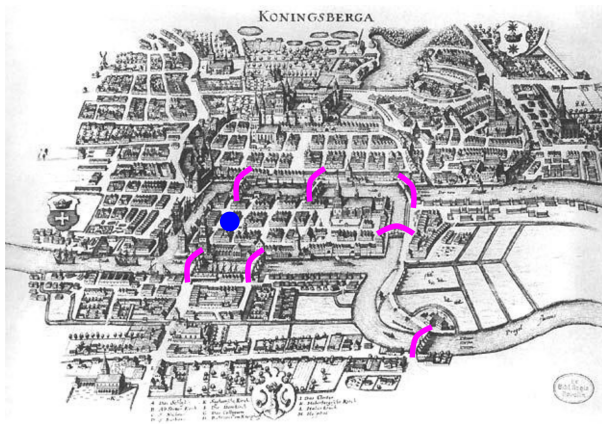
Souris puisque c'est graphe

Bruno Teheux

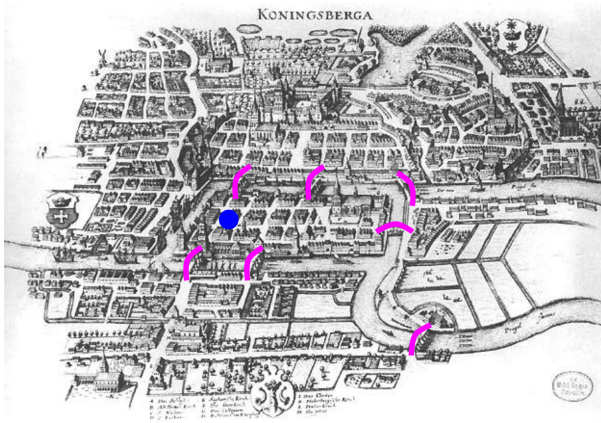
Université du Luxembourg

Un beau jour de 1735, à Königsberg

Un beau jour de 1735, à Königsberg



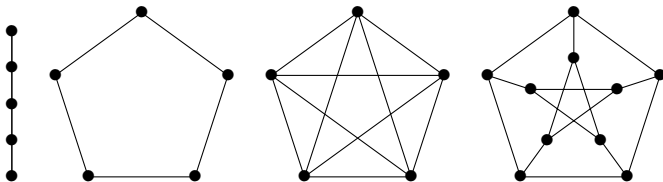
Un beau jour de 1735, à Königsberg



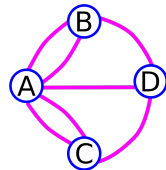
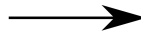
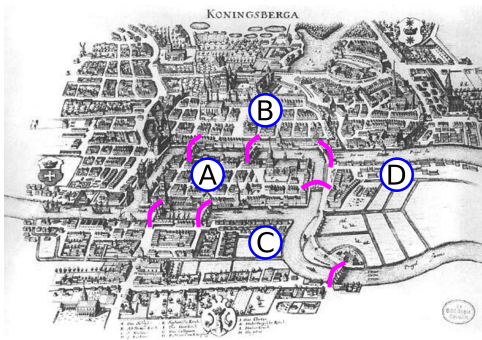
Partir de ●, parcourir tous les ponts une seule fois, revenir en ●.

Et Euler inventa les graphes

Un graphe est donné par des sommets dont certains sont reliés par des arrêtes.



Königsberg a son graphe



Les graphes considérés par Euler

Un graphe est *touristique* s'il peut être parcouru par

- un circuit fermé (*i.e.* qui revient au sommet initial)
- qui passe par chaque arête du graphe une et une seule fois.

Un tel parcours est un parcours *touristique*.

Les graphes considérés par Euler

Un graphe est *touristique* s'il peut être parcouru par

- un circuit fermé (*i.e.* qui revient au sommet initial)
- qui passe par chaque arête du graphe une et une seule fois.

Un tel parcours est un parcours *touristique*.

Exemples.



est touristique



n'est pas touristique



est touristique ?

Le problème considéré par Euler

Comment caractériser les graphes touristiques ?

Atelier. [5 minutes]

Identifier un maximum de graphes touristiques.

Comment reconnaître si un graphe est touristique ?

Rappel. Un graphe est *touristique* s'il peut être parcouru par

- un circuit fermé (*i.e.* qui revient au sommet initial)
- qui passe par chaque arrête du graphe une et une seule fois.

Le problème considéré par Euler

Comment caractériser les graphes touristiques ?

Atelier. [5 minutes]

Identifier un maximum de graphes touristiques.

Comment reconnaître si un graphe est touristique ?

Rappel. Un graphe est *touristique* s'il peut être parcouru par

- un circuit fermé (*i.e.* qui revient au sommet initial)
- qui passe par chaque arrête du graphe une et une seule fois.

Vocabulaire. On convient qu'un *parcours* ne passe pas deux fois par le même sommet.

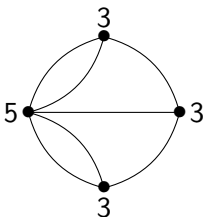
$E(G)$ est l'ensemble des arrêtes de G .

Et Euler caractérisa les graphes touristiques

Le *dégré* d'un sommet est le nombre d'arêtes qui lui sont adjacentes.

Théorème Un graphe connexe est touristique si et seulement si tous ses sommets ont un degré pair.

Le graphe de Königsberg n'est pas touristique.

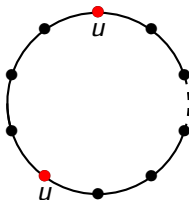


Une solution locale pour un problème global ...

Il existe une preuve constructive

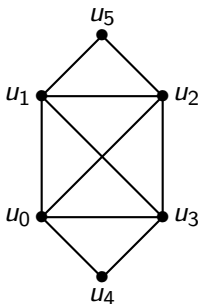
Théorème. Un graphe connexe est touristique si et seulement si tous ses sommets ont un degré pair.

Preuve. \Rightarrow Supposons que T soit un parcours fermé de G qui passe par chaque arête une et une seule fois.

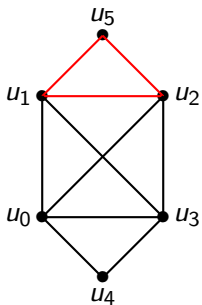


Chaque apparition d'un sommet u est accompagnée d'une arête entrant dans u et d'une arête sortant de u .

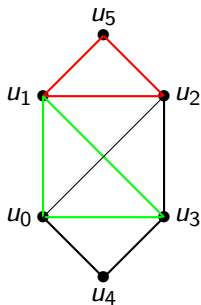
\Leftarrow Construction d'un parcours touristique comme une union de parcours fermés disjoints (algorithme de Hierholzer).



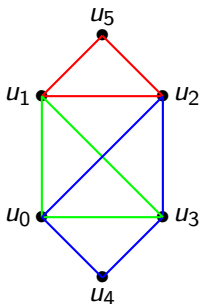
⇐ Construction d'un parcours touristique comme une union de parcours fermés disjoints (algorithme de Hierholzer).



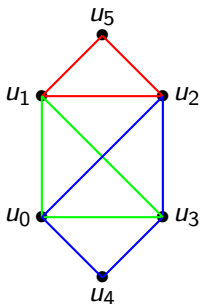
⇐ Construction d'un parcours touristique comme une union de parcours fermés disjoints (algorithme de Hierholzer).



⇐ Construction d'un parcours touristique comme une union de parcours fermés disjoints (algorithme de Hierholzer).



⇐ Construction d'un parcours touristique comme une union de parcours fermés disjoints (algorithme de Hierholzer).



Parcours touristique :

u_1 u_5 u_2 u_2 u_3 u_1 u_0 u_3 u_4 u_0 u_2 u_1

⇐ Construction d'un parcours touristique.

```
1: function Tourist( $G, u$ )
2:   Construire un parcours fermé  $C = u_1 \cdots u_\ell$  à partir de  $u$ 
3:    $G' \leftarrow G - E(C)$ 
4:   while  $\exists u_k \in C$  tel que  $\deg_{G'}(u_k) \geq 2$  do
5:     Construire dans  $G'$  un parcours fermé  $C'$  à partir de  $u_k$ 
6:      $C \leftarrow u_1 \cdots u_{k-1} C' u_{k+1} \cdots u_\ell$ 
7:      $G' \leftarrow G' - E(C')$ 
8:   return  $C$ 
```

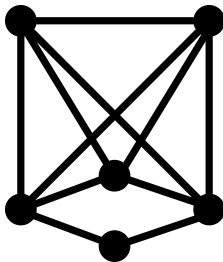
L'algorithme est correct car

Ligne 2 : un tel parcours existe car $\deg(u) > 0$ est pair

Ligne 5 : chaque sommet de $G - E(C)$ a un degré pair

Ligne 8 : par connexité, un parcours touristique a été trouvé

Exercice. Appliquer l'algorithme de Hierholzer pour trouver un parcours touristique de



Une variation non fermée

Un graphe est *semi-touristique* s'il possède un parcours **non-fermé** qui passe par chaque arrête du graphe.

Une variation non fermée

Un graphe est *semi-touristique* s'il possède un parcours **non-fermé** qui passe par chaque arrête du graphe.

Théorème. Un graphe connexe est semi-touristique si et seulement si il possède exactement deux sommets de degré impair.

Une variation non fermée

Un graphe est *semi-touristique* s'il possède un parcours **non-fermé** qui passe par chaque arrête du graphe.

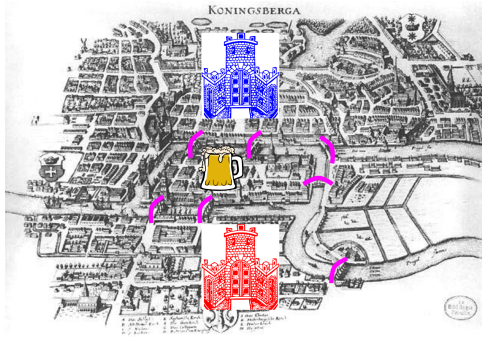
Théorème. Un graphe connexe est semi-touristique si et seulement si il possède exactement deux sommets de degré impair.

Preuve. \implies Ajouter une nouvelle arrête entre le sommet initial et le sommet final d'un parcours semi-touristique : on obtient un parcours touristique.

\impliedby Ajouter une arrête entre les deux sommets de degré impair : tous les sommets ont alors un degré pair.

Le graphe de Königsberg n'est pas semi-touristique.

Pour poursuivre la réflexion



Où le chevalier bleu doit-il construire un pont pour

- qu'il puisse se rendre de son château à la taverne en passant par chaque pont une et une seule fois,
- que le chevalier rouge n'ait pas cette possibilité.

[...]

Le problème du postier

La carte d'une ville peut se représenter par un graphe dont
les sommets sont les intersections des rues
les arrêtes sont les rues

Le problème du Postier. Concevoir une « tournée » dans le graphe qui passe par toutes les arrêtes au moins une fois et de longueur minimum.

- Si le graphe est touristique, tout parcours touristique est une solution
- Si le graphe est semi-touristique [...]
- Sinon [...]

Dans la littérature

Un parcours touristique est appelé *cycle eulérien*.

Un graphe touristique est appelé graphe *eulérien* .

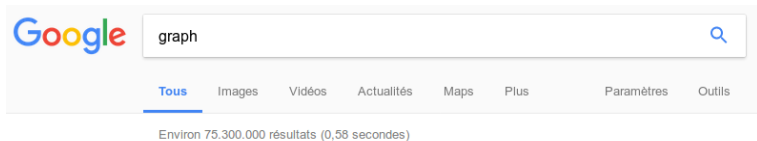
Au coeur d'un moteur de recherche

Le graphe de recommandation de Google

Le problème de Google

Le nombre de pages web fouillées par Google est

130 000 000 000 000.



Comment fait Google pour vous recommander une page ?

Google attribue un score à chaque page

Google attribue un score (*page rank*) à chaque page.

Le score correspond à la « réputation » de la page.

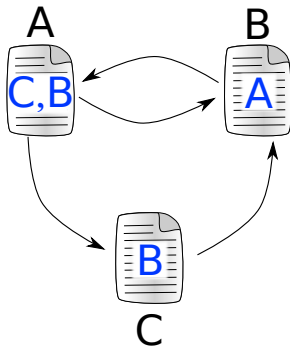
Les pages qui ont un score élevé seront recommandées en premier.

Comment ces scores sont-ils calculés ?

Internet est un graphe

Le graphe d'internet est dirigé :

- les sommets sont les pages web,
- une arrête relie A à B si A contient un lien vers B .



Les scores sont calculés *à l'intérieur d'internet*

Plus une page est référencée par d'autres pages, meilleur est son score.

Les scores sont calculés *à l'intérieur d'internet*

Plus une page est référencée par d'autres pages, meilleur est son score.

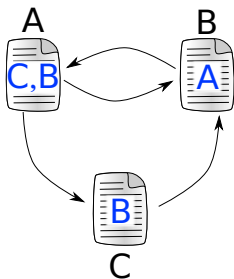
Deux règles son respectées :

- ⊕ on accorde un meilleur score à des pages référencées par des pages qui ont elles-même un bon score,
- ⊖ on accorde moins d'importance aux liens qui proviennent de pages qui contiennent de nombreux liens.

En pratique, le calcul des scores

Désignons par x_A, x_B, x_C les scores des pages A, B, C .

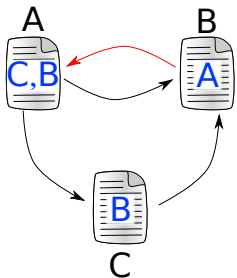
- ⊕ on accorde un meilleur score à des pages référencées par des pages qui ont elles-mêmes un bon score



En pratique, le calcul des scores

Désignons par x_A, x_B, x_C les scores des pages A, B, C .

- ⊕ on accorde un meilleur score à des pages référencées par des pages qui ont elles-mêmes un bon score

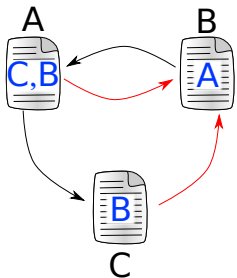


$$x_A = x_B$$

En pratique, le calcul des scores

Désignons par x_A, x_B, x_C les scores des pages A, B, C .

- ⊕ on accorde un meilleur score à des pages référencées par des pages qui ont elles-mêmes un bon score



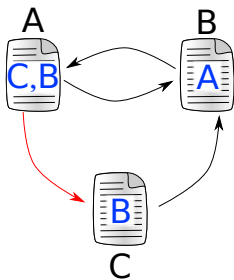
$$x_A = x_B$$

$$x_B = x_A + x_C$$

En pratique, le calcul des scores

Désignons par x_A, x_B, x_C les scores des pages A, B, C.

⊕ on accorde un meilleur score à des pages référencées par des pages qui ont elles-mêmes un bon score



$$x_A = x_B$$

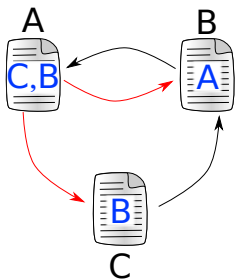
$$x_B = x_A + x_C$$

$$x_C = x_A$$

En pratique, le calcul des scores

Désignons par x_A, x_B, x_C les scores des pages A, B, C.

- ⊖ on accorde moins d'importance aux liens qui proviennent de pages qui contiennent de nombreux liens.



$$x_A = x_B$$

$$x_B = \frac{x_A}{2} + x_C$$

$$x_C = \frac{x_A}{2}$$

Calculer les scores, c'est résoudre un système d'équations

$$\left\{ \begin{array}{l} x_A = x_B \\ x_B = \frac{x_A}{2} + x_C \\ x_C = \frac{x_A}{2} \end{array} \right.$$

Calculer les scores, c'est résoudre un système d'équations

On ajoute une condition de normalisation

$$\begin{cases} x_A = x_B \\ x_B = \frac{x_A}{2} + x_C \\ x_C = \frac{x_A}{2} \\ x_A + x_B + x_C = 100 \end{cases}$$

A vous de jouer !

Calculer les scores, c'est résoudre un système d'équations

On ajoute une condition de normalisation

$$\begin{cases} x_A = x_B \\ x_B = \frac{x_A}{2} + x_C \\ x_C = \frac{x_A}{2} \\ x_A + x_B + x_C = 100 \end{cases}$$

A vous de jouer !

On trouve l'unique solution

$$x_A = 40, \quad x_B = 40, \quad x_C = 20$$

Donc les pages *A* et *B* seront classées avant la page *C*.

Dans la vraie vie, les choses sont plus compliquées

Le système d'équation a une taille énorme

Pour assurer l'existence et l'unicité de la solution, le modèle est légèrement adapté.

On peut se servir du même modèle pour calculer d'autres classements (sportifs par exemple).

Dans la vraie vie, les choses sont plus compliquées

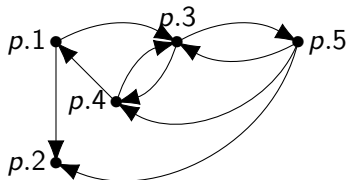
Le système d'équation a une taille énorme

Pour assurer l'existence et l'unicité de la solution, le modèle est légèrement adapté.

On peut se servir du même modèle pour calculer d'autres classements (sportifs par exemple).

Le problème sous forme matricielle

En appliquant les règles \oplus et \ominus au graphe



on obtient le système $A\mathbf{x} = \mathbf{x}$ où $\mathbf{x} = (x_1, \dots, x_5)^T$ et

$$A = \begin{pmatrix} 0 & 0 & 0 & 1/2 & 0 \\ 1/2 & 0 & 0 & 0 & 1/3 \\ 1/2 & 0 & 0 & 1/2 & 1/3 \\ 0 & 0 & 1/2 & 0 & 1/3 \\ 0 & 0 & 1/2 & 0 & 0 \end{pmatrix}.$$

Un tel système n'a pas nécessairement de solution non nulle unique.

Vers un classement unique

L'idée de L. Page et S. Bring est de transformer A en deux étapes.

- On consruit B en remplaçant les colonnes nulles de A par des colonnes de $1/n$ (où n est le nombre de pages)

$$B = \begin{pmatrix} 0 & 1/5 & 0 & 1/2 & 0 \\ 1/2 & 1/5 & 0 & 0 & 1/3 \\ 1/2 & 1/5 & 0 & 1/2 & 1/3 \\ 0 & 1/5 & 1/2 & 0 & 1/3 \\ 0 & 1/5 & 1/2 & 0 & 0 \end{pmatrix}.$$

Vers un classement unique

L'idée de L. Page et S. Bring est de transformer A en deux étapes.

- On construit B en remplaçant les colonnes nulles de A par des colonnes de $1/n$ (où n est le nombre de pages)

$$B = \begin{pmatrix} 0 & 1/5 & 0 & 1/2 & 0 \\ 1/2 & 1/5 & 0 & 0 & 1/3 \\ 1/2 & 1/5 & 0 & 1/2 & 1/3 \\ 0 & 1/5 & 1/2 & 0 & 1/3 \\ 0 & 1/5 & 1/2 & 0 & 0 \end{pmatrix}.$$

- On construit C en multipliant tout élément de B par $\alpha = 0.85$ et en ajoutant $(1 - \alpha)/n$:

$$C = 0.85 \cdot B + \frac{0.15}{n} \cdot J$$

où J est la matrice qui a des 1 partout.

Le théorème de Perron – Frobenius

La matrice C vérifie les hypothèses du thm. de Perron – Frobenius.

Ainsi,

- À un multiple près, il n'existe qu'une solution \mathbf{x}_0 de $C\mathbf{x} = \mathbf{x}$.
- Les éléments de \mathbf{x}_0 sont positifs

Le théorème de Perron – Frobenius

La matrice C vérifie les hypothèses du thm. de Perron – Frobenius.

Ainsi,

- À un multiple près, il n'existe qu'une solution \mathbf{x}_0 de $C\mathbf{x} = \mathbf{x}$.
- Les éléments de \mathbf{x}_0 sont positifs
- Les colonnes de C^k convergent vers \mathbf{x}_0 quand k tend vers l'infini
- L'unicité de la solution est obtenue en ajoutant la condition de normalisation.

Sur notre exemple...

Voici les valeurs des éléments de la deuxième colonne de C^k

n	c_1	c_2	c_3	c_4	c_5
1	0.2000	0.2000	0.2000	0.2000	0.2000
2	0.1490	0.2057	0.2907	0.2057	0.1490
3	0.1524	0.1705	0.2579	0.2307	0.1885
4	0.1570	0.1772	0.2752	0.2220	0.1686
5	0.1545	0.1746	0.2690	0.2248	0.1771
6	0.1552	0.1755	0.2711	0.2242	0.1740
7	0.1551	0.1751	0.2704	0.2243	0.1750
8	0.1551	0.1753	0.2706	0.2243	0.1747
9	0.1551	0.1752	0.2705	0.2243	0.1748

Donc les pages seront présentées dans l'ordre

3 4 2 5 1

Le comportement limite d'un surfeur aléatoire

Imaginons un surfeur qui navigue le web au hasard.

Toutes les unités de temps, il suit un lien sur la page qu'il est en train de lire.

Le comportement limite d'un surfeur aléatoire

S'il est sur la page 1, la probabilité qu'il aboutisse sur la page 2 en **une** unité de temps est

$$A_{12}$$

Le comportement limite d'un surfeur aléatoire

S'il est sur la page 1, la probabilité qu'il aboutisse sur la page 2 en **une** unité de temps est

$$A_{12}$$

S'il est sur la page 1, la probabilité qu'il aboutisse sur la page 2 en **deux** unités de temps est

$$A_{11}A_{12} + A_{12}A_{22} + A_{13}A_{32} + A_{14}A_{42} + A_{15}A_{52}$$

c'est-à-dire $(A^2)_{12}$

Le comportement limite d'un surfeur aléatoire

S'il est sur la page 1, la probabilité qu'il aboutisse sur la page 2 en **une** unité de temps est

$$A_{12}$$

S'il est sur la page 1, la probabilité qu'il aboutisse sur la page 2 en **deux** unités de temps est

$$A_{11}A_{12} + A_{12}A_{22} + A_{13}A_{32} + A_{14}A_{42} + A_{15}A_{52}$$

c'est-à-dire $(A^2)_{12}$

Proposition. S'il est sur la page i , la probabilité qu'il aboutisse sur la page j en **k** unités de temps est

$$(A^k)_{ij}$$

La suite des A^k n'a pas forcément de comportement limite

Imaginons un surfeur qui navigue le web au hasard.

Toutes les unités de temps,

- il suit 85 fois sur 100 un lien sur la page sur laquelle il se trouve,
- il ouvre une nouvelle page au hasard 15 fois sur 100.

La suite des A^k n'a pas forcément de comportement limite

Imaginons un surfeur qui navigue le web au hasard.

Toutes les unités de temps,

- il suit 85 fois sur 100 un lien sur la page sur laquelle il se trouve,
- il ouvre une nouvelle page au hasard 15 fois sur 100.

Proposition. S'il est sur la page i , la probabilité qu'il aboutisse sur la page j en k unités de temps est

$$(C^k)_{ij}$$

Le comportement limite d'un surfeur aléatoire

Proposition. Si c est une colonne de la limite C' des C^k alors c_j est la probabilité que le surfeur se retrouve *in fine* sur la page j .

À la limite, la position d'un surfeur dans le graphe ne dépend pas de la condition initiale.

Calcul du plus court chemin

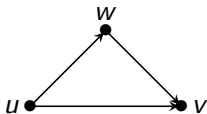
Le problème du GPS

Qu'est-ce qu'un graphe dirigé ?

Définition. Un *graphe dirigé* $G = (V, E)$ est la donnée d'un ensemble fini $V \neq \emptyset$ de *sommets* et d'un ensemble $E \subseteq V \times V$ d'arêtes (ou *arc*) dirigé.

Un arc (u, v) a une *origine* u et une *extrémité* v .

Exemple. $G = (\{u, v, w\}, \{(u, v), (u, w), (w, v)\})$



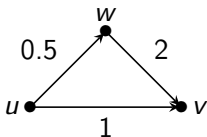
Qu'est-ce qu'un graphe dirigé à poids ?

Définition. Un *graphe dirigé à poids* $G = (V, E, p)$ est donné par une fonction $p: E \rightarrow \mathbb{R}^+$ sur les arcs d'un graphe dirigé $G = (V, E)$.

Qu'est-ce qu'un graphe dirigé à poids ?

Définition. Un *graphe dirigé à poids* $G = (V, E, p)$ est donné par une fonction $p: E \rightarrow \mathbb{R}^+$ sur les arcs d'un graphe dirigé $G = (V, E)$.

Exemple.



$$p(u, v) = 1, \quad p(u, w) = 0.5, \quad p(w, v) = 2.$$

Le problème du plus court chemin

- Graphe dirigé simple à poids $G = (V, E, p)$.

Le problème du plus court chemin

- Graphe dirigé simple à poids $G = (V, E, p)$.
- Le poids $p(C)$ d'une chaîne $C = (u, e_1, v_1, \dots, v_{t-1}, e_t, v)$ est définie par $p(C) = \sum_{i=1}^t p(e_i)$

Definition. Un *plus court chemin* entre $u, v \in V$ est une chaîne C qui a un poids minimum parmi les chaînes entre u et v .

Remarques.

- S'il n'y a pas de chaînes de u à v , on pose $d(u, v) = \infty$
- La fonction p s'étend à $V \times V$ en posant $p((u, v)) = \infty$ si $(u, v) \notin E$.
- On fixe $u \in V$.

Le problème du plus court chemin

- Graphe dirigé simple à poids $G = (V, E, p)$.
- Le poids $p(C)$ d'une chaîne $C = (u, e_1, v_1, \dots, v_{t-1}, e_t, v)$ est définie par $p(C) = \sum_{i=1}^t p(e_i)$

Definition. Un *plus court chemin* entre $u, v \in V$ est une chaîne C qui a un poids minimum parmi les chaînes entre u et v .

Remarques.

- S'il n'y a pas de chaînes de u à v , on pose $d(u, v) = \infty$
- La fonction p s'étend à $V \times V$ en posant $p((u, v)) = \infty$ si $(u, v) \notin E$.
- On fixe $u \in V$.

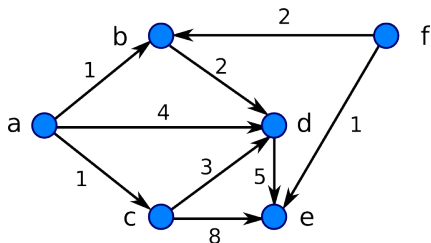
On fournit un algorithme qui produit un plus court chemin entre u et n'importe quel élément de $v \in V$ (si ce chemin existe).

Dijkstra's algorithm (1959)

```
1:  $X \leftarrow \{u\}$ 
2: for  $v \in V$  do
3:    $T(v) \leftarrow p(u, v)$ 
4:    $C_v \leftarrow (u, v)$ 
5: end for
6: while  $X \neq V$  do
7:   Choose  $v \in V \setminus X$  such that  $\forall y \in V \setminus X, T(v) \leq T(y)$ 
8:    $X \leftarrow X \cup \{v\}$ 
9:   for  $y \in V \setminus X$  do
10:    if  $T(y) > T(v) + p(v, y)$  then
11:       $T(y) \leftarrow T(v) + p(v, y)$ 
12:       $C(y) \leftarrow [C(v), y]$ 
13:    end if
14:  end for
15: end while
```

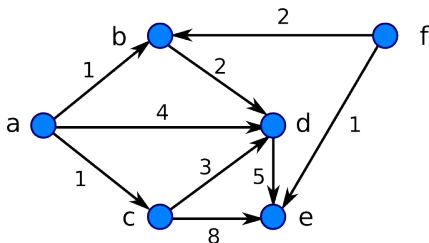
Exemple

Appliquer l'algorithme de Dijkstra au graphe (choix de $u = a$).



Exemple

Appliquer l'algorithme de Dijkstra au graphe (choix de $u = a$).



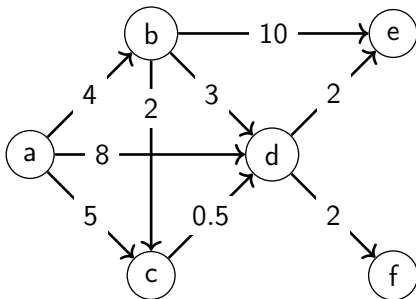
Proposition (L'algorithme de Dijkstra est correct)

À chaque étape de l'algorithme

1. pour tout $v \in X$, $T(v)$ est le poids minimal des chaînes de u à v .
2. pour tout $w \notin X$, $T(w)$ est le poids minimal des chaînes de u à w dont tous les sommets sauf w sont dans X .

Exercice

Appliquer l'algorithme de Dijkstra avec *a* comme sommet initial.



Moi ? Ici ?

Raisons historiques

Raisons géographiques

Offre en mathématiques :

- Math.en.Jeans
- Scienteens Lab (venez nous voir)
- Exposition
- Exposés
- Bachelor + Master

<http://math.uni.lu/outreach>