Graphes et langages (I)
Introduction aux graphes

bruno.colombel@univ-amu.fr

DUT Informatique
IUT d'Aix-Marseille
Site d'Arles

2018 — 2019

De quoi s'agit-il ?

Types de graphes

- graphes orientés
- Graphes non-orientés
- Graphes pondérés

Les problèmes classiques

Degré, chemin, circuit, cycle

Graphe planaire et graphe complet

Représentations des graphes

Parcours

- Parcours en largeur
- Parcours en profondeur

Sommaire

- La théorie des graphes est une théorie *informatique* et *mathématique*

- La théorie des graphes est une théorie *informatique* et *mathématique*
- Applications

- La théorie des graphes est une théorie *informatique* et *mathématique*
- Applications
  - dans tous les domaines liés à la notion de réseau

- La théorie des graphes est une théorie *informatique* et *mathématique*
- Applications
  - dans tous les domaines liés à la notion de réseau
  - dans l'imagerie numérique

- La théorie des graphes est une théorie *informatique* et *mathématique*
- Applications
  - dans tous les domaines liés à la notion de réseau
  - dans l'imagerie numérique
  - architecture des ordinateurs

- La théorie des graphes est une théorie *informatique* et *mathématique*
- Applications
  - dans tous les domaines liés à la notion de réseau
  - dans l'imagerie numérique
  - architecture des ordinateurs
  - langages

- La théorie des graphes est une théorie *informatique* et *mathématique*
- Applications
  - dans tous les domaines liés à la notion de réseau
  - dans l'imagerie numérique
  - architecture des ordinateurs
  - langages
  - langages compilés

- La théorie des graphes est une théorie *informatique* et *mathématique*
- Applications
  - dans tous les domaines liés à la notion de réseau
  - dans l'imagerie numérique
  - architecture des ordinateurs
  - langages
  - langages compilés
  - expressions régulières

- La théorie des graphes est une théorie *informatique* et *mathématique*
- Applications
  - dans tous les domaines liés à la notion de réseau
  - dans l'imagerie numérique
  - architecture des ordinateurs
  - langages
  - langages compilés
  - expressions régulières

La résolution des problèmes posés par cette théorie s'élabore essentiellement à l'aide d'algorithmes

Métro de Marseille

Sommaire

Graphes orientés

Définition

Un *graphe orienté* $\mathscr{G}$ est constitué :

Graphes orientés

Définition

Un *graphe orienté* $\mathscr{G}$ est constitué :

- d'un ensemble fini $\mathcal{S}$ (les sommets)

Graphes orientés

Définition

Un *graphe orienté* $\mathscr{G}$ est constitué :

- d'un ensemble fini $\mathcal{S}$ (les sommets)
- d'un ensemble fini $\mathcal{A}$ (les arêtes)

Graphes orientés

Définition

Un *graphe orienté* $\mathscr{G}$ est constitué :

- d'un ensemble fini $\mathcal{S}$ (les sommets)
- d'un ensemble fini $\mathcal{A}$ (les arêtes)
- d'une application $\delta : \mathcal{A} \to \mathcal{S}^2$ qui à une arrête associe 2 sommets

Graphes orientés

Définition

Un *graphe orienté* $\mathscr{G}$ est constitué :

- d'un ensemble fini $\mathcal{S}$ (les sommets)
- d'un ensemble fini $\mathcal{A}$ (les arêtes)
- d'une application $\delta : \mathcal{A} \to \mathcal{S}^2$ qui à une arrête associe 2 sommets

Un graphe $\mathscr{G}$ est désigné par le triplet :

$$\mathscr{G} = (\mathcal{S}; \mathcal{A}; \delta)$$

Graphes orientés

Soit $\mathscr{G} = (\mathcal{S}; \mathcal{A}; \delta)$ un graphe orienté.

Vobulaire – Notation

- $|\mathcal{S}|$: *ordre* du graphe (nombre de sommets)

Graphes orientés

Soit $\mathscr{G} = (\mathcal{S}; \mathcal{A}; \delta)$ un graphe orienté.

Vobulaire – Notation

- $|\mathcal{S}|$: *ordre* du graphe (nombre de sommets)
- Si $\delta(\varepsilon) = (s, t)$ alors :
  - s est le *début* de l'arête ε
  - t est le *fin* de l'arête ε
  - s et t sont les extrémités de ε.

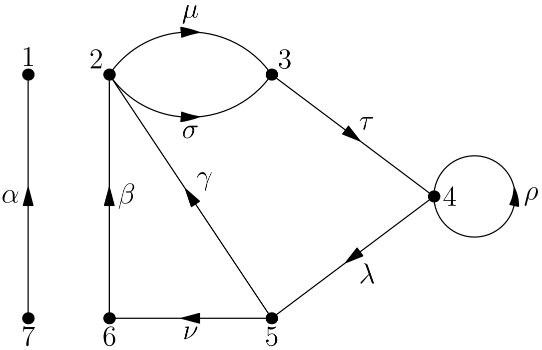
Graphes orientés

Soit $\mathscr{G} = (\mathcal{S}; \mathcal{A}; \delta)$ un graphe orienté.

Vobulaire – Notation

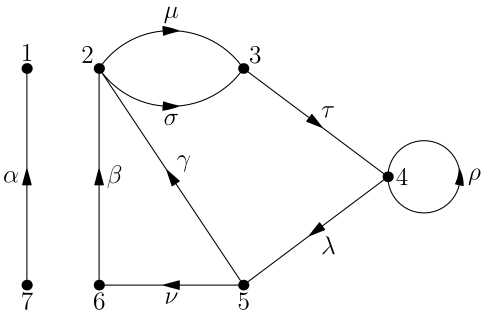
- $|\mathcal{S}|$: *ordre* du graphe (nombre de sommets)
- Si $\delta(\varepsilon) = (s, t)$ alors :
  - s est le *début* de l'arête ε
  - t est le *fin* de l'arête ε
  - s et t sont les extrémités de ε.
- deux sommets s et t sont *adjacents* s'il existe une arête entre s et t

Graphes orientés



arêtes	extrémités
α	$(7, 1)$
β	$(6, 2)$
γ	$(5, 2)$
λ	$(4, 5)$
μ	$(2, 3)$
ν	$(5, 6)$
ρ	$(4, 4)$
σ	$(2, 3)$
τ	$(3, 4)$

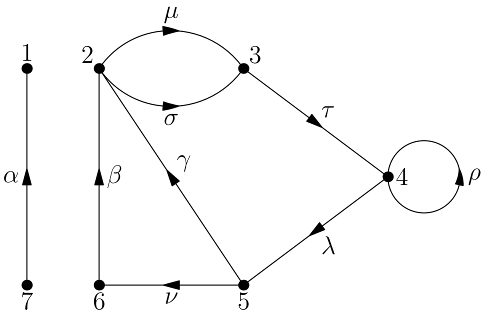
Graphes orientés



Vobulaire – Notation

- *arêtes parallèles* : même début et même fin

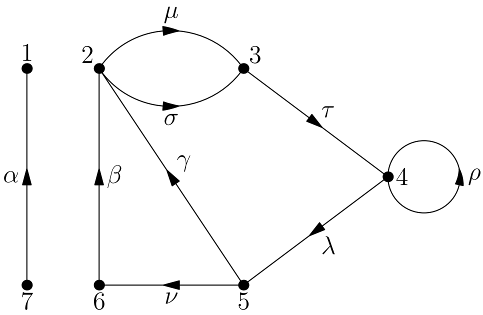
Graphes orientés



Vobulaire – Notation

- *arêtes parallèles* : même début et même fin
- *boucles* : arête dont le début et la fin coïncident

Graphes orientés



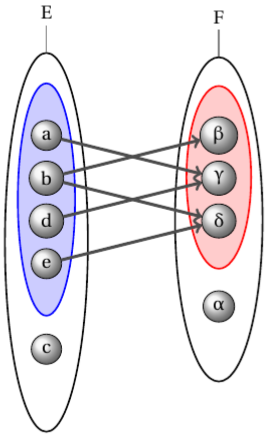
Vobulaire – Notation

- *arêtes parallèles* : même début et même fin
- *boucles* : arête dont le début et la fin coïncident

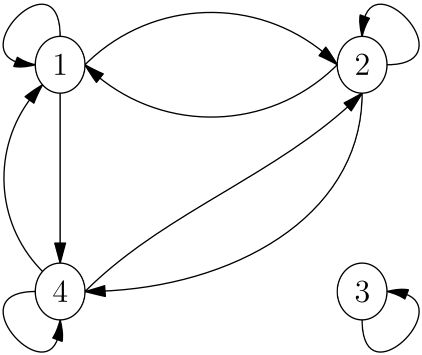
Définition

Un graphe est *simple* quand il ne comporte ni arêtes parallèles ni boucle.

Exemple (Diagramme sagittal d'une relation à 2 ensembles)



Exemple (Diagramme sagittal d'une relation à 1 ensemble)



ATTENTION ! ! !

1. Un graphe n'est pas un dessin. C'est un objet abstrait constitué de deux ensembles et d'une application.

ATTENTION ! ! !

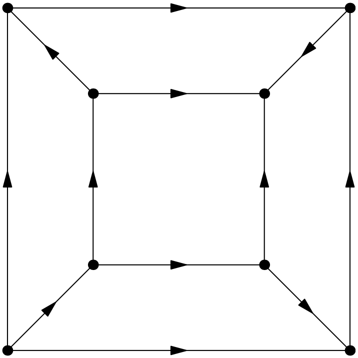
1. Un graphe n'est pas un dessin. C'est un objet abstrait constitué de deux ensembles et d'une application.
2. Représenter un graphe peut aider à mieux le comprendre à condition que le résultat ne soit pas trop compliqué

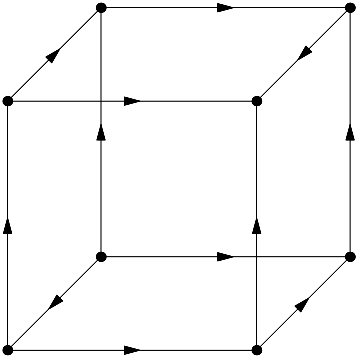
ATTENTION ! ! !

1. Un graphe n'est pas un dessin. C'est un objet abstrait constitué de deux ensembles et d'une application.
2. Représenter un graphe peut aider à mieux le comprendre à condition que le résultat ne soit pas trop compliqué
3. Il existe une multitude de façon de représenter un graphe

Graphes orientés

Les deux dessins ci-dessous représentent le même graphe :





Graphes non-orientés

Définition

Un *graphe non-orienté* $\mathscr{G}$ est constitué :

Graphes non-orientés

Définition

Un *graphe non-orienté* $\mathscr{G}$ est constitué :

- d'un ensemble fini $\mathcal{S}$ (les sommets)

Graphes non-orientés

Définition

Un *graphe non-orienté* $\mathscr{G}$ est constitué :

- d'un ensemble fini $\mathcal{S}$ (les sommets)
- d'un ensemble fini $\mathcal{A}$ (les arêtes)

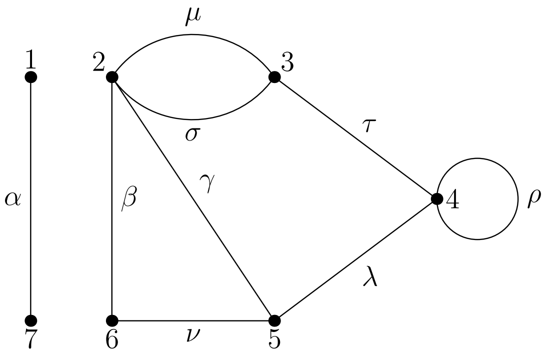
Graphes non-orientés

Définition

Un *graphe non-orienté* $\mathscr{G}$ est constitué :

- d'un ensemble fini $\mathcal{S}$ (les sommets)
- d'un ensemble fini $\mathcal{A}$ (les arêtes)
- d'une application $\delta : \mathcal{A} \to \mathcal{S}_1 \cup \mathcal{S}_2$ où $\mathcal{S}_i$ est l'ensemble des parties à k éléments de $\mathcal{S}$.

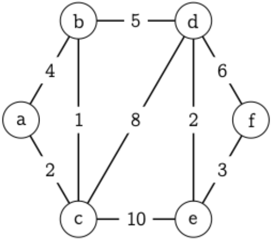
Graphes non-orientés



arêtes	extrémités
α	$\{7,1\}$
β	$\{6,2\}$
γ	$\{5,2\}$
λ	$\{4,5\}$
μ	$\{2,3\}$
ν	$\{5,6\}$
ρ	$\{4\}$
σ	$\{2,3\}$
τ	$\{3,4\}$

Graphes pondérés

C'est un graphe dont les arêtes sont affectées d'un « poids ».



Graphes pondérés

Définition

Un graphe pondéré est un quadruplet $\mathscr{G} = (\mathcal{S}; \mathcal{A}; \delta; p)$ où :

- $(\mathcal{S}; \mathcal{A}; \delta)$ est un graphe ;

Graphes pondérés

Définition

Un graphe pondéré est un quadruplet $\mathscr{G} = (\mathcal{S}; \mathcal{A}; \delta; p)$ où :

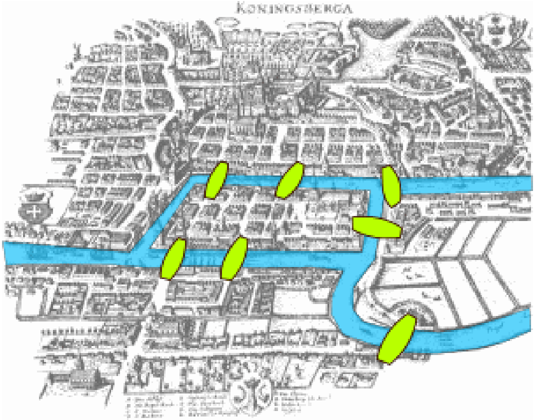
- $(\mathcal{S}; \mathcal{A}; \delta)$ est un graphe ;
- p est une fonction dite de *poids*

$$p : \mathcal{A} \to \mathbb{R}$$

Sommaire

Pont de Königsberg

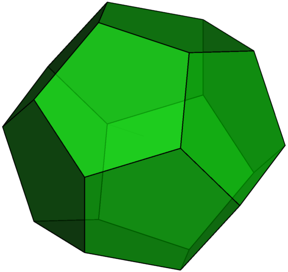
Deux îles sont reliées au reste de la ville par 7 ponts :



Existe-t-il un chemin empruntant tous les ponts une fois et une seule ?

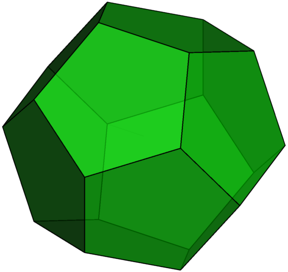
Dodécaèdre d'Hamilton

- Le jeu commercialisé par Hamilton était constitué d'un dodécaèdre



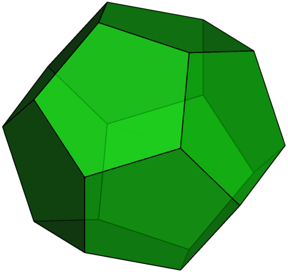
Dodécaèdre d'Hamilton

- Le jeu commercialisé par Hamilton était constitué d'un dodécaèdre
- Chaque sommet portait le nom d'une ville



Dodécaèdre d'Hamilton

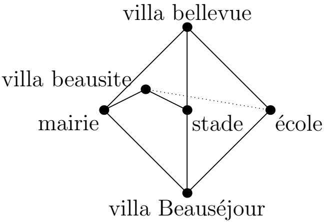
- Le jeu commercialisé par Hamilton était constitué d'un dodécaèdre
- Chaque sommet portait le nom d'une ville



- Le but était de se déplacer de ville en ville le long des arêtes en passant une fois et une seule par chacune des ville en revenant au point de départ

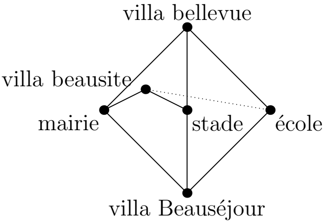
Problème des trois villas

- Les propriétaires de 3 villas veulent tous être reliés à la mairie, au stade et à l'école par des voies privées



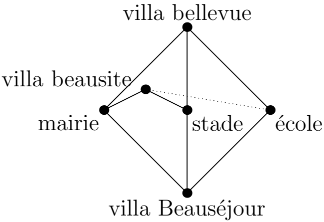
Problème des trois villas

- Les propriétaires de 3 villas veulent tous être reliés à la mairie, au stade et à l'école par des voies privées
- Ils ne se supportent pas et refusent carrefours, ponts ou souterrains



Problème des trois villas

- Les propriétaires de 3 villas veulent tous être reliés à la mairie, au stade et à l'école par des voies privées
- Ils ne se supportent pas et refusent carrefours, ponts ou souterrains



Personne n'a encore trouvé de solution !

Problème de quatre couleurs

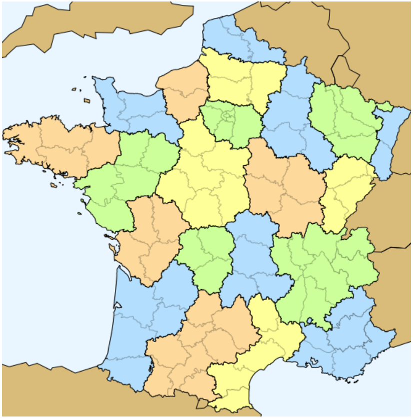
- Quand on colorie une carte de géographie, on essaie d'utiliser des couleurs différentes aux parties qui se touchent.

Problème de quatre couleurs

- Quand on colorie une carte de géographie, on essaie d'utiliser des couleurs différentes aux parties qui se touchent.
- Vers 1880, Gunthrie remarqua qu'on y arrivait toujours avec 4 couleurs

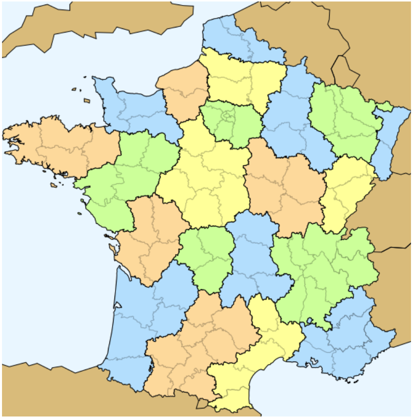
Problème de quatre couleurs

- Quand on colorie une carte de géographie, on essaie d'utiliser des couleurs différentes aux parties qui se touchent.
- Vers 1880, Gunthrie remarqua qu'on y arrivait toujours avec 4 couleurs



Problème de quatre couleurs

- Quand on colorie une carte de géographie, on essaie d'utiliser des couleurs différentes aux parties qui se touchent.
- Vers 1880, Gunthrie remarqua qu'on y arrivait toujours avec 4 couleurs

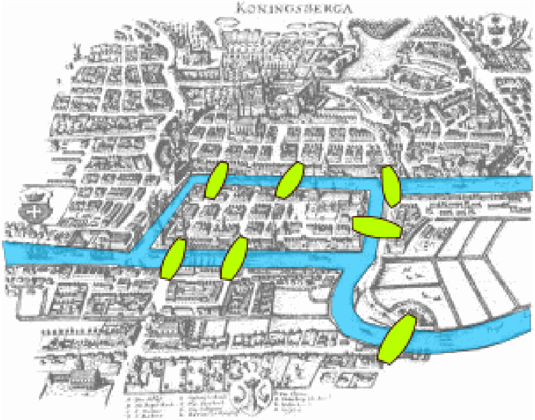


Est-ce toujours le cas ?

Sommaire

Pont de Königsberg

Deux îles sont reliés au reste de la ville par 7 ponts :



Existe-t-il un chemin empruntant tous les ponts une fois et une seule ?

Degré d'un sommet

Définition

Soit s le sommet d'un graphe non-orienté. Le *degré* de s est égal au nombre d'arêtes dont s est une extrémité.

Degré d'un sommet

Définition

Soit s le sommet d'un graphe non-orienté. Le *degré* de s est égal au nombre d'arêtes dont s est une extrémité.

Attention, les boucles comptent double !

Degré d'un sommet

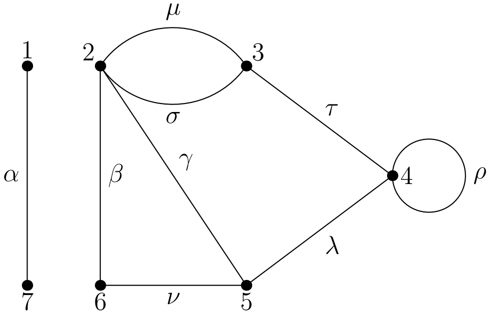
Définition

Soit s le sommet d'un graphe non-orienté. Le *degré* de s est égal au nombre d'arêtes dont s est une extrémité.

Attention, les boucles comptent double !

Exemple

$$d(1) = 1$$
$$d(2) = 4$$
$$d(4) = 4$$



Degré d'un sommet

Définition

Soit s le sommet d'un graphe orienté.

Degré d'un sommet

Définition

Soit s le sommet d'un graphe orienté.

- le *degré sortant* $d^+(s)$ est le nombre d'arêtes dont s est le début

Degré d'un sommet

Définition

Soit s le sommet d'un graphe orienté.

- le *degré sortant* $d^{+}(s)$ est le nombre d'arêtes dont s est le début
- le *degré entrant* $d^{-}(s)$ est le nombre d'arêtes dont s est la fin

Degré d'un sommet

Définition

Soit s le sommet d'un graphe orienté.

- le *degré sortant* $d^+(s)$ est le nombre d'arêtes dont s est le début
- le *degré entrant* $d^-(s)$ est le nombre d'arêtes dont s est la fin

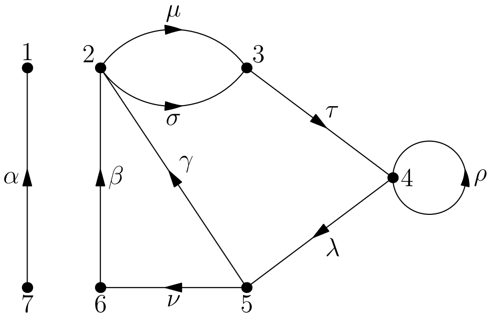
En effaçant le sens de parcours des arêtes, on obtient un graphe non-orienté. On a alors :

$$d(s) = d^+(s) + d^-(s)$$

Degré d'un sommet

Exemple

$$d^+(4) = 2$$
$$d^-(4) = 2$$
$$d(4) = 2 + 2 = 4$$



Chemin de longueur n

Définition

1. Soient s et t de sommets d'un graphe orienté $\mathscr{G} = (\mathcal{S}; \mathcal{A}; \delta)$. La suite $(S_0 = s, \varepsilon_1, S_1, \varepsilon_2, \cdots, \varepsilon_n, S_n = t)$ est un *chemin de longueur n* menant de s à t si :

$$\delta(\varepsilon_i) = S_{i-1}S_i$$

Chemin de longueur n

Définition

1. Soient s et t de sommets d'un graphe orienté $\mathscr{G} = (\mathcal{S}; \mathcal{A}; \delta)$. La suite $(S_0 = s, \varepsilon_1, S_1, \varepsilon_2, \cdots, \varepsilon_n, S_n = t)$ est un *chemin de longueur n* menant de s à t si :

$$\delta(\varepsilon_i) = S_{i-1}S_i$$

Autrement dit, on peut aller du sommet s au sommet t en suivant les arêtes du graphe et leur sens de parcours en n étapes.

Chemin de longueur n

Définition

1. Soient s et t de sommets d'un graphe orienté $\mathscr{G} = (\mathcal{S}; \mathcal{A}; \delta)$. La suite $(S_0 = s, \varepsilon_1, S_1, \varepsilon_2, \cdots, \varepsilon_n, S_n = t)$ est un *chemin de longueur n* menant de s à t si :

$$\delta(\varepsilon_i) = S_{i-1}S_i$$

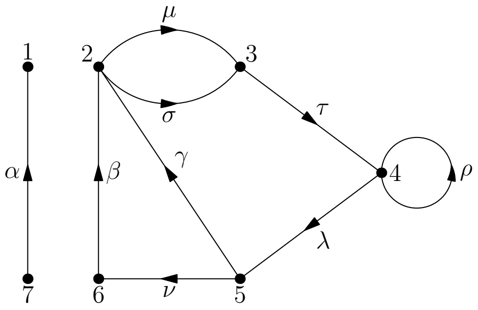
   Autrement dit, on peut aller du sommet s au sommet t en suivant les arêtes du graphe et leur sens de parcours en n étapes.

2. Un chemin dont le départ et l'arrivée coïncident est un *circuit*

Chemin de longueur n

Exemple

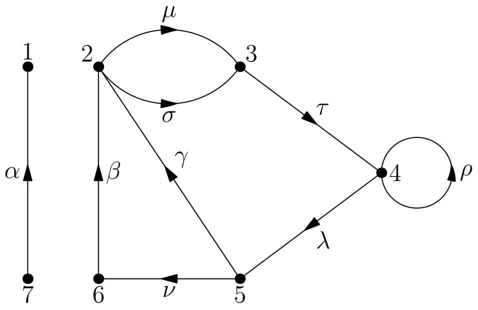
- $(5; \gamma; 2; \sigma; 3; \tau; 4)$ est un chemin de longueur 3 entre les sommets 5 et 4



Chemin de longueur n

Exemple

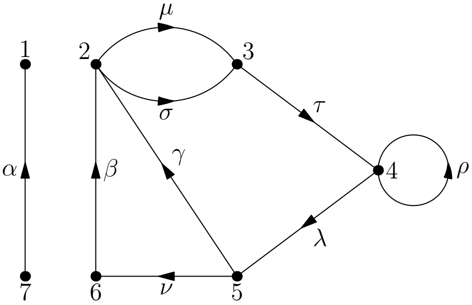
- $(5; \gamma; 2; \sigma; 3; \tau; 4)$ est un chemin de longueur 3 entre les sommets 5 et 4
- $(3; \tau; 4; \lambda; 5; \nu; 6; \beta; 2)$ est un chemin de longueur 4 entre les sommets 3 et 2



Chemin de longueur n

Exemple

- $(5; \gamma; 2; \sigma; 3; \tau; 4)$ est un chemin de longueur 3 entre les sommets 5 et 4
- $(3; \tau; 4; \lambda; 5; \nu; 6; \beta; 2)$ est un chemin de longueur 4 entre les sommets 3 et 2



Remarque

La définition est analogue pour les graphes non-orientés

Graphe connexe

Définition

1. Un graphe non-orienté est dit *connexe* s'il existe un chemin entre deux sommets quelconque de ce graphe

Graphe connexe

Définition

1. Un graphe non-orienté est dit *connexe* s'il existe un chemin entre deux sommets quelconque de ce graphe
2. Un graphe orienté est dit *fortement connexe* s'il existe un chemin entre deux sommets quelconque de ce graphe

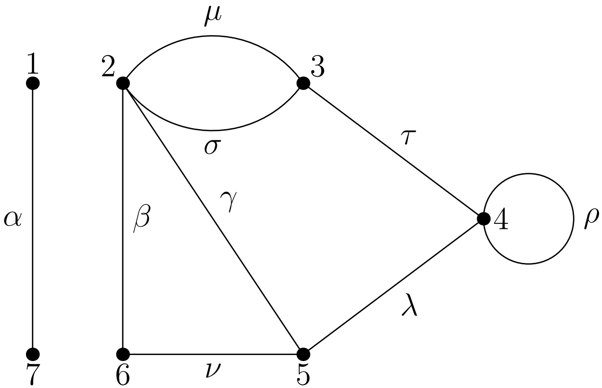
Graphe connexe

Définition

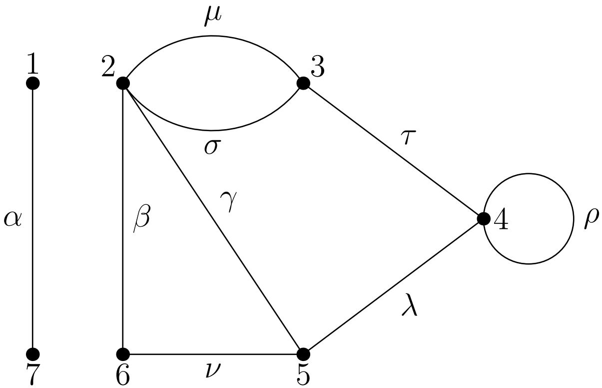
1. Un graphe non-orienté est dit *connexe* s'il existe un chemin entre deux sommets quelconque de ce graphe
2. Un graphe orienté est dit *fortement connexe* s'il existe un chemin entre deux sommets quelconque de ce graphe

Pour les graphes orientés, le sens de parcours des arêtes doit être respecté

Graphe connexe

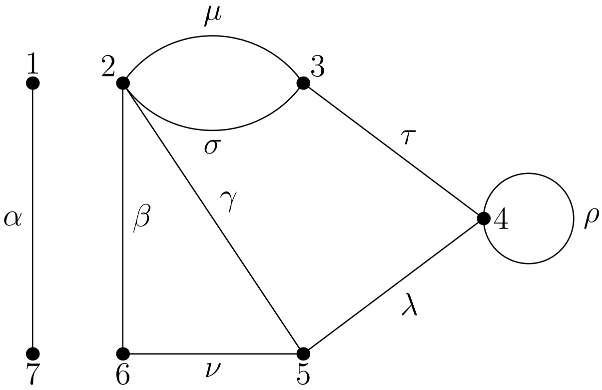


Graphe connexe



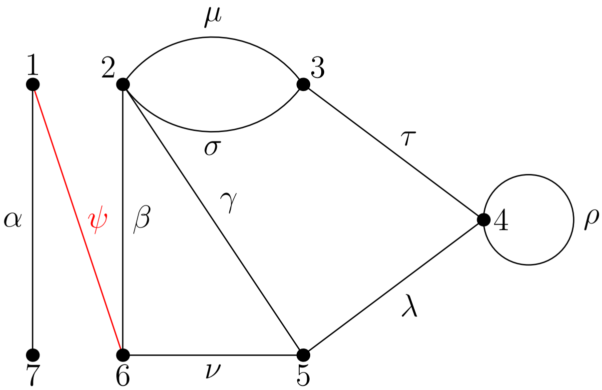
- $\mathscr{G}$ n'est pas connexe : on ne peut pas *passer* du sommet 1 au sommet 5

Graphe connexe



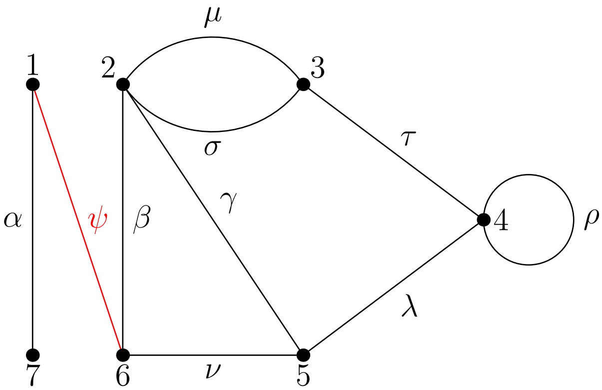
- $\mathscr{G}$ n'est pas connexe : on ne peut pas *passer* du sommet 1 au sommet 5
- On dit que $\mathscr{G}$ a deux *composantes connexes*

Graphe connexe



On ajoute l'arête $6 \to 1$

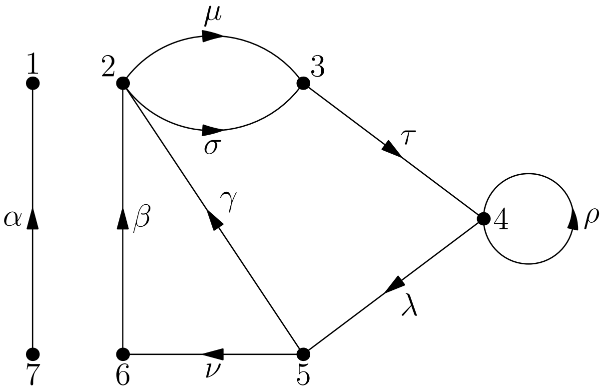
Graphe connexe



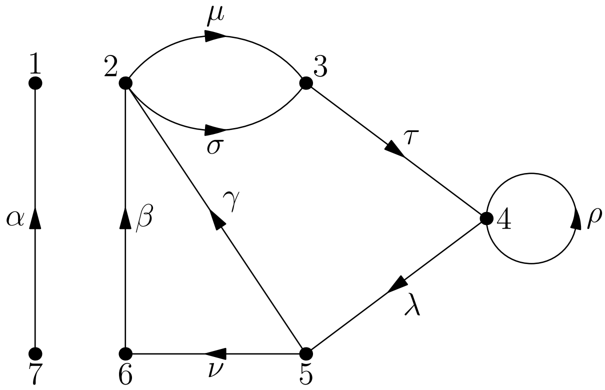
On ajoute l'arête 6 $\to$ 1

$\mathscr{G}$ est maintenant connexe

Graphe connexe

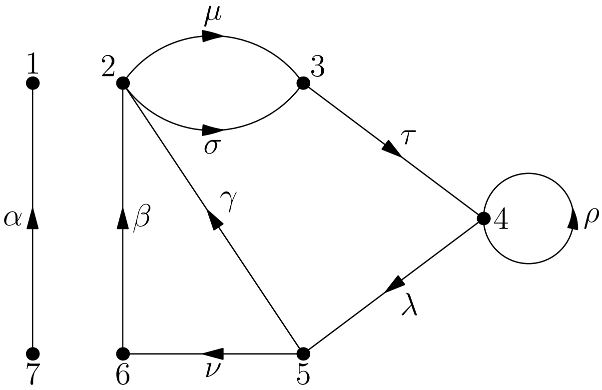


Graphe connexe



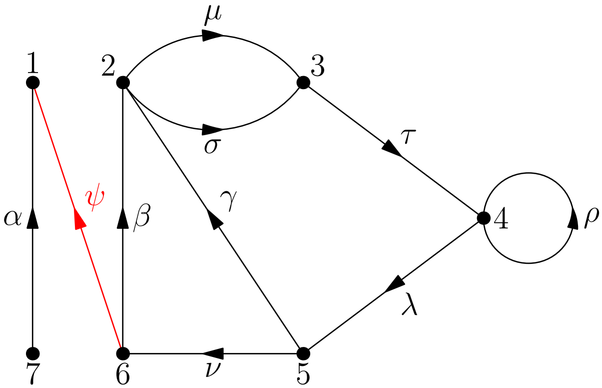
- $\mathscr{G}$ n'est pas fortement connexe : on ne peut pas *passer* du sommet 1 au sommet 5

Graphe connexe



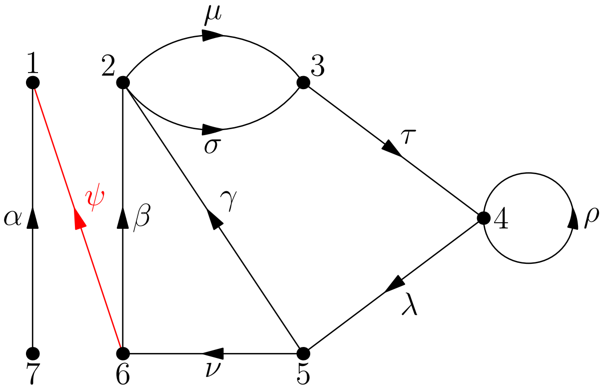
- $\mathscr{G}$ n'est pas fortement connexe : on ne peut pas *passer* du sommet 1 au sommet 5
- On dit que $\mathscr{G}$ a deux *composantes connexes*

Graphe connexe



On ajoute l'arête $6 \rightarrow 1$

Graphe connexe

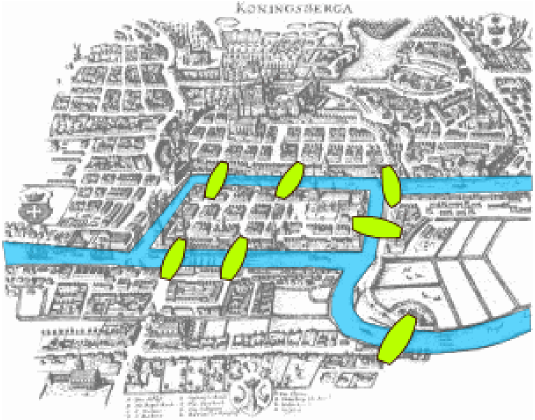


On ajoute l'arête $6 \to 1$

$\mathscr{G}$ n'est pas fortement connexe : on ne peut (toujours) pas *passer* du sommet 1 au sommet 5

Pont de Königsberg

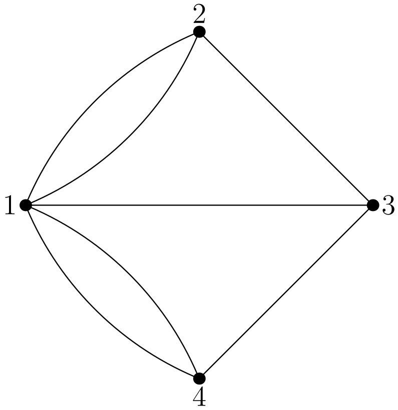
Deux îles sont reliées au reste de la ville par 7 ponts :



Existe-t-il un chemin empruntant tous les ponts une fois et une seule ?

Pont de Königsberg

Modélisons la ville par un graphe



Cela revient à chercher une chaine ou un circuit qui passe par toutes les arête et tous les sommets sont utiliser 2 fois la même arête

Chaine eulérienne

Définition

1. Une chaîne est dite *eulérienne* si :
   - Elle contient toutes les arêtes du graphes
   - Chaque arête n'est « décrite » qu'une seule fois.

Chaine eulérienne

Définition

1. Une chaîne est dite *eulérienne* si :
   - Elle contient toutes les arêtes du graphes
   - Chaque arête n'est « décrite » qu'une seule fois.
2. Un circuit eulérien est une chaine eulérienne fermée

Chaîne eulérienne

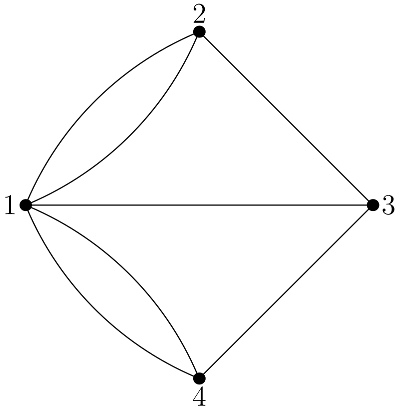
Théorème (Euler)

1. *$\mathscr{G}$ étant un graphe connexe, les deux propriétés suivantes sont équivalentes :*
   - *Tous les sommets de $\mathscr{G}$ sont de degré pair*
   - *$\mathscr{G}$ admet un cycle eulérien*

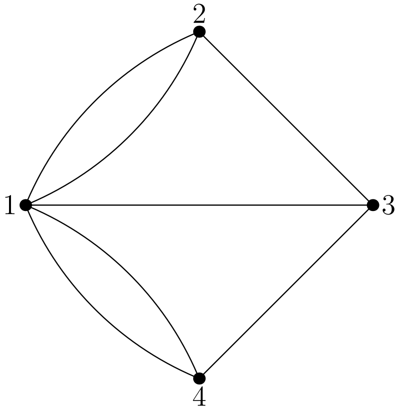
Théorème (Euler)

1. $\mathscr{G}$ *étant un graphe connexe, les deux propriétés suivantes sont équivalentes :*
   - *Tous les sommets de* $\mathscr{G}$ *sont de degré pair*
   - $\mathscr{G}$ *admet un cycle eulérien*
2. $\mathscr{G}$ *étant un graphe connexe, les deux propriétés suivantes sont équivalentes :*
   - *Deux sommets (et deux seulement)* A *et* B *de* $\mathscr{G}$ *sont de degré impair*
   - $\mathscr{G}$ *admet une chaîne eulérienne d'extrémité* A *et* B

Pont de Königsberg



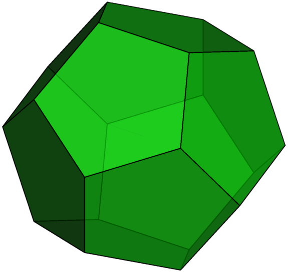
Pont de Königsberg



$d(2) = d(4) = 3$ et $d(1) = 5$
Pas de chaine eulérienne ! ! !

Dodécaèdre d'Hamilton

- Le jeu commercialisé par Hamilton était constitué d'un dodécaèdre
- Chaque sommet portait le nom d'une ville



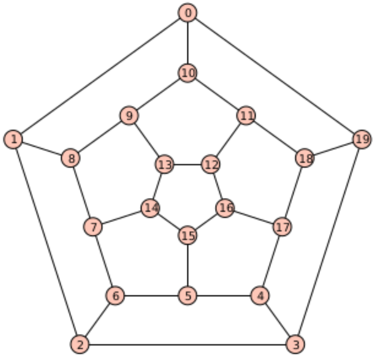
- Le but était de se déplacer de ville en ville le long des arêtes en passant une fois et une seule par chacune des ville en revenant au point de départ

Graphe hamiltonien

Modélisons la situation par une graphe

Graphe hamiltonien

Modélisons la situation par une graphe



Graphe hamiltonien

Définition

1. Un cycle hamiltonien d'un graphe non-orienté est une chaîne qui passe par tous les sommets du graphe une, et une seule fois avant de revenir au sommet de départ

Graphe hamiltonien

Définition

1. Un cycle hamiltonien d'un graphe non-orienté est une chaîne qui passe par tous les sommets du graphe une, et une seule fois avant de revenir au sommet de départ
2. Un graphe qui contient un cycle hamiltonien est appelé un graphe hamiltonien

Graphe hamiltonien

Malheureusement, il n'existe aucune propriété générale permettant de conclure si un graphe est hamiltonien ou non.

Graphe hamiltonien

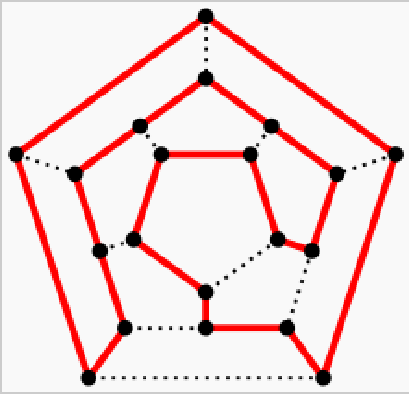
Malheureusement, il n'existe aucune propriété générale permettant de conclure si un graphe est hamiltonien ou non.

La question de trouver un cycle hamiltonien minimal est connu sous le nom de :

problème du voyageur de commerce

Il s'agit d'un des thèmes de Recherche Opérationnelle

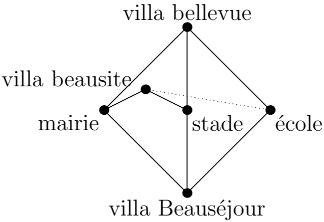
Graphe hamiltonien



Sommaire

Problème des trois villas

- Les propriétaires de 3 villas veulent tous être reliés à la mairie, au stade et à l'école par des voies privées
- Ils ne se supportent pas et refusent carrefours, ponts ou souterrains



Personne n'a encore trouvé de solution !

Graphe planaire

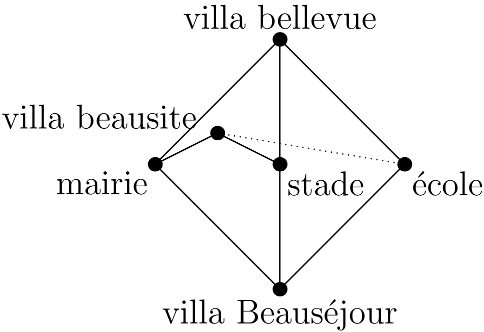
Définition

Un graphe est *planaire* s'il peut être dessiné dans un plan sans que deux arêtes quelconques se coupent

Autrement dit, il est possible de tracer ses arêtes les unes à la suite des autres sans recouper celles déjà tracées

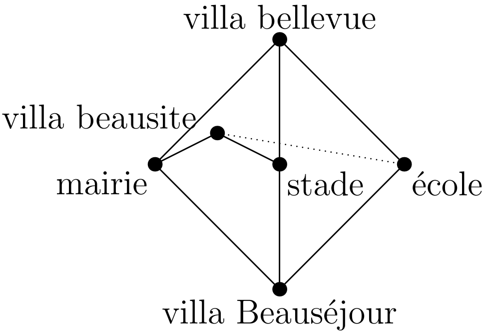
Graphe planaire

Le problème des trois villas revient à savoir si le graphe est planaire ou non



Graphe planaire

Le problème des trois villas revient à savoir si le graphe est planaire ou non

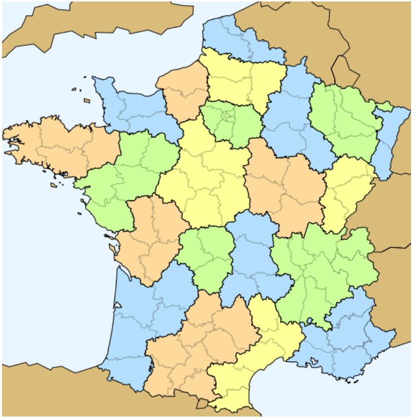


Ce graphe n'est pas planaire

Le problème est insoluble

Problème de quatre couleurs

- Quand on colorie une carte de géographie, on essaie d'utiliser des couleurs différentes aux parties qui se touchent.
- Vers 1880, Gunthrie remarqua qu'on y arrivait toujours avec 4 couleurs



Est-ce toujours le cas ?

Graphe complet

Définition

1. Un sous-graphe $\mathscr{G}'$ d'un graphe $\mathscr{G}$ est un graphe composé de certains sommets de $\mathscr{G}$ ainsi que de toutes les arêtes qui relient ces sommets.
2. Un sous-graphe stable d'un graphe $\mathscr{G}$ est un sous-graphe dont aucune arête ne relie les sommets.

Graphe complet

Définition

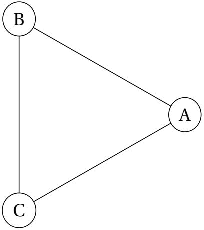
Un graphe est dit *complet* si tous les sommets de ce graphes sont adjacents (2 à 2).

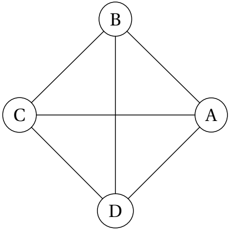
Graphe complet

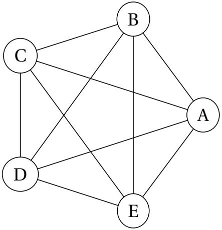
Définition

Un graphe est dit *complet* si tous les sommets de ce graphes sont adjacents (2 à 2).

Exemple







Nombre chromatique

Définition

1. *Colorer* un graphe consiste à affecter une couleur à chacun des sommets de sorte que deux sommets adjacents ne portent pas la même couleur.

Nombre chromatique

Définition

1. *Colorer* un graphe consiste à affecter une couleur à chacun des sommets de sorte que deux sommets adjacents ne portent pas la même couleur.
2. Le nombre chromatique d'un graphe est le plus petit nombre de couleurs permettent de le colorer.
   Il est souvent noté $\gamma(\mathscr{G})$

Nombre chromatique

Théorème

Le nombre chromatique d'un graphe complet d'ordre n est égal à n.

Nombre chromatique

Théorème

Le nombre chromatique d'un graphe complet d'ordre n est égal à n.

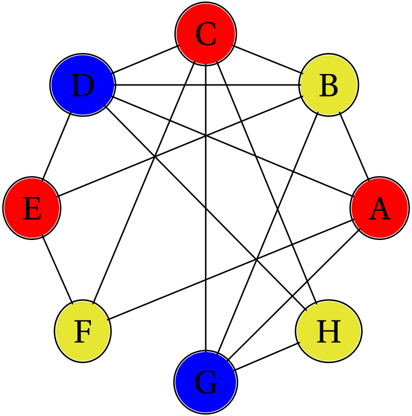
Conséquences :

Si un graphe admet un sous graphe complet d'ordre p, alors le nombre chromatique de ce graphe est supérieur ou égal à p.

Nombre chromatique

Exemple

1. Le graphe ci-contre contient un triangle complet (A — B — D par Ex). Donc, $\gamma(G) \geq 3$.

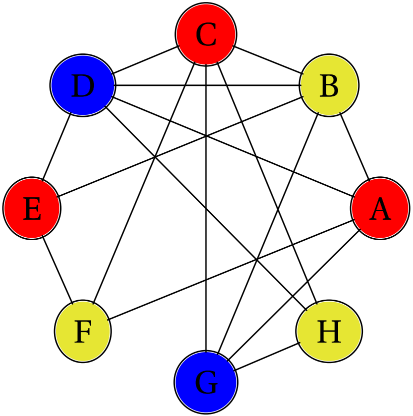


Nombre chromatique

Exemple

1. Le graphe ci-contre contient un triangle complet (A — B — D par Ex). Donc, $\gamma(G) \geq 3$.
2. On doit colorer le graphe pour conclure : on peut colorer le graphe avec 3 couleurs seulement donc :

$$\gamma(G) = 3$$



Algorithme de Welsch-Powell

Entrées : Graphe G avec ses sommets
Sorties : Les sommets affectés d'une couleur
début

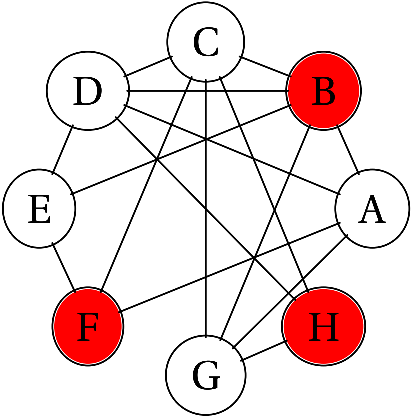
> Trier les sommets de G par ordre de degré décroissant ;
> **tant que** *Tous les sommets de G ne sont pas colorés* **faire**
>
> > Affecter à color une nouvelle couleur;
> > Colorer avec color le 1^er^ sommet du tableau non encore coloré ;
> > Colorer avec color tous les sommets de G non encore colorés et non adjacents à un sommet coloré avec color ;
>
> **fin**
> Afficher les sommets et leurs couleurs

fin

Algorithme de Welsch-Powell

Sommets	B	C	D	A	F	E	G	H
Degrés	5	5	5	4	3	3	3	3

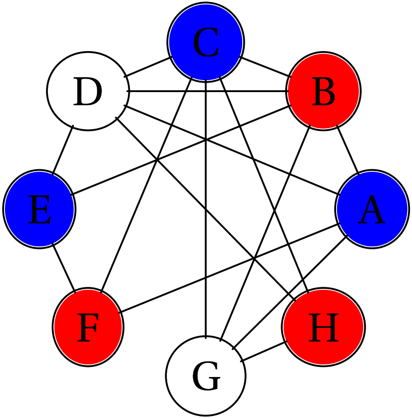
On choisit une couleur pour le premier sommet de la liste : On choisit de colorer le sommet B en rouge.
On colorie en rouge les sommets non adjacents à B et non adjacents entre eux.



Algorithme de Welsch-Powell

Sommets	B	C	D	A	F	E	G	H
Degrés	5	5	5	4	3	3	3	3

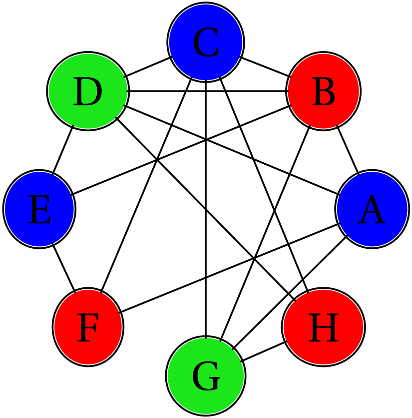
On réitère la procédé : on colorie C en bleu.
On colorie A et E en bleu.



Algorithme de Welsch-Powell

Sommets	B	C	D	A	F	E	G	H
Degrés	5	5	5	4	3	3	3	3

On réitère le procédé : on colorie en vert le point D.
On colorie ensuite G en vert.



Théorème des quatre couleurs

Théorème

Tout graphe planaire peut être colorié avec au plus quatre couleurs

Sommaire

Matrice d'adjacence

Soit $\mathscr{G}$ un graphe orienté d'ordre n (n sommets). On suppose les sommets numérotés de 1 à n. On note S_1, S_2, ..., S_n ces sommets.

Définition

La matrice d'adjacence de $\mathscr{G}$ est la matrice $A \in \mathcal{M}_n(\mathbb{R})$ où :

$$a_{i,j} = \text{nombre d'arête de début } S_i \text{et de fin } S_j$$

Matrice d'adjacence

Soit $\mathscr{G}$ un graphe orienté d'ordre n (n sommets). On suppose les sommets numérotés de 1 à n. On note S_1, S_2, ..., S_n ces sommets.

Définition

La matrice d'adjacence de $\mathscr{G}$ est la matrice $A \in \mathcal{M}_n(\mathbb{R})$ où :

$$a_{i,j} = \text{nombre d'arête de début } S_i \text{et de fin } S_j$$

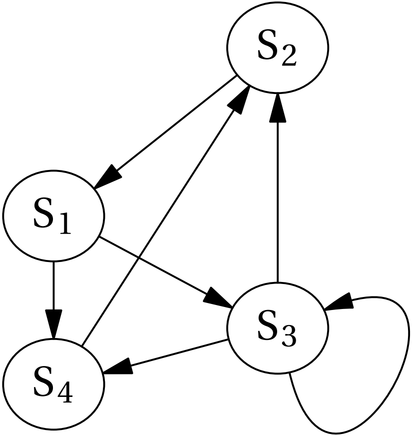
Remarque

La matrice A dépend de l'ordre dans le lequel on énumère les sommets

Matrice d'adjacence

Exemple

$$A = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$



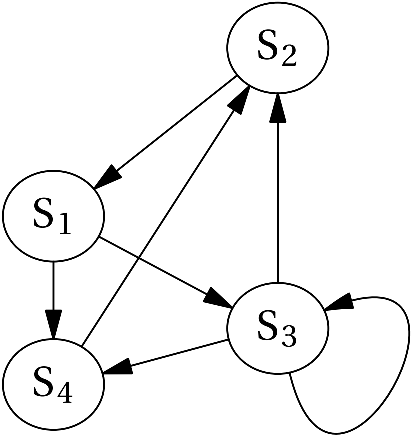
Matrice d'adjacence

Théorème

Dans la matrice A^p, le coefficient situé à l'intersection de la ligne i et de la colonne j est égal au nombre de chaînes de longueur p partant du sommet numéro i at arrivant au sommet numéro j.

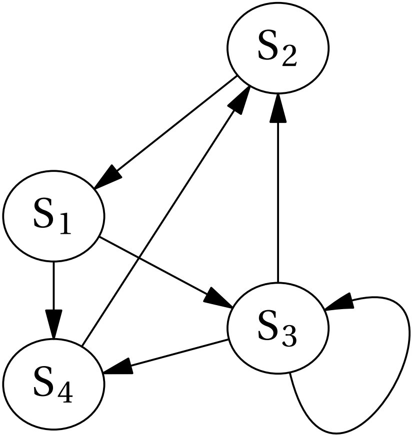
Matrice d'adjacence

Exemple



Exemple

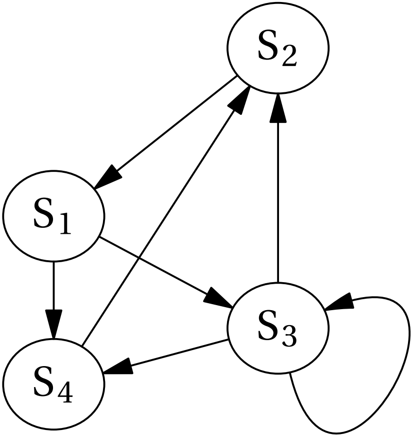
$$A^{10} = \begin{pmatrix} 46 & 74 & 63 & 63 \\ 26 & 46 & 37 & 37 \\ 60 & 100 & 86 & 86 \\ 14 & 26 & 23 & 23 \end{pmatrix}$$



Matrice d'adjacence

Exemple

$$A^{10} = \begin{pmatrix} 46 & 74 & 63 & 63 \\ 26 & 46 & 37 & 37 \\ 60 & 100 & 86 & 86 \\ 14 & 26 & 23 & 23 \end{pmatrix}$$

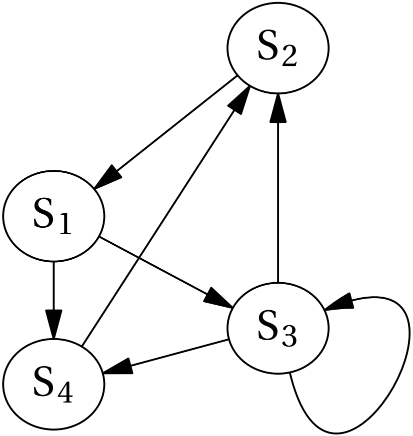


- 37 chaînes orientées de S_2 vers S_4 (car $a_{24} = 37$).

Matrice d'adjacence

Exemple

$$A^{10} = \begin{pmatrix} 46 & 74 & 63 & 63 \\ 26 & 46 & 37 & 37 \\ 60 & 100 & 86 & 86 \\ 14 & 26 & 23 & 23 \end{pmatrix}$$



- 37 chaînes orientées de S_2 vers S_4 (car $a_{24} = 37$).
- 100 chaînes orientées de S_3 vers S_2 (car $a_{32} = 37$).

Matrice d'adjacence

Soit $\mathscr{G}$ un graphe non-orienté d'ordre n (n sommets). On suppose les sommets numérotés de 1 à n. On note S_1, S_2, ..., S_n ces sommets.

Définition

la matrice d'adjacence de $\mathscr{G}$ est la matrice $A \in \mathcal{M}_n(\mathbb{R})$ définie par :

$$a_{i,j} = \text{nombres d'arêtes entre les sommets } S_i \text{ et } S_j$$

Matrice d'adjacence

Soit $\mathscr{G}$ un graphe non-orienté d'ordre n (n sommets). On suppose les sommets numérotés de 1 à n. On note S_1, S_2, ..., S_n ces sommets.

Définition

la matrice d'adjacence de $\mathscr{G}$ est la matrice $A \in \mathcal{M}_n(\mathbb{R})$ définie par :

$$a_{i,j} = \text{nombres d'arêtes entre les sommets } S_i \text{ et } S_j$$

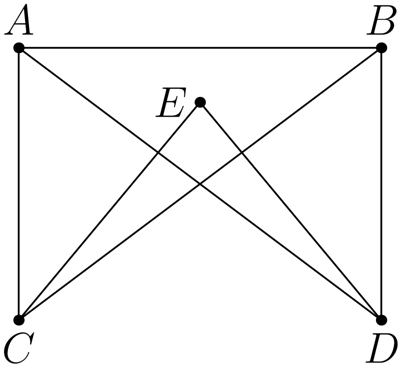
la matrice d'adjacence d'un graphe non-orienté est donc symétrique

Matrice d'adjacence

Exemple

En choisissant comme ordre des sommets l'ordre alphabétique :

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

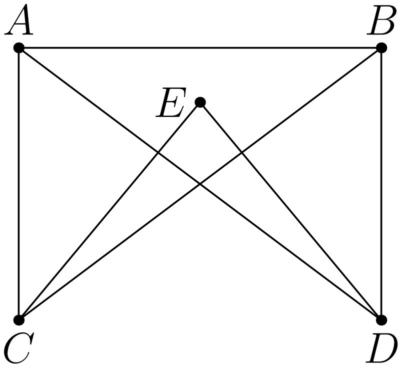


Matrice d'adjacence

Exemple

En choisissant comme ordre des sommets l'ordre alphabétique :

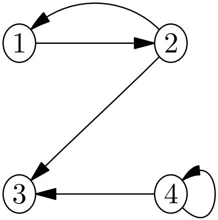
$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$



La matrice est bien symétrique

Autres représentations

Il existe d'autres représentations



représentation sous forme de **listes** :

$$\begin{array}{rcl} S & = & \{1, 2, 3, 4\}, \\ A & = & \{(1, 2), (2, 1), (2, 3), (4, 3), (4, 4)\} \end{array}$$

ou d'une **liste unique** (*comment lire cette liste ?*) :

$$L = ({\color{red}4}, {\color{orange}1, 2, 3, 4}, 1, 2, 2, 1, 2, 3, 4, 3, 4, 4)$$

Graphes pondérés

Définition

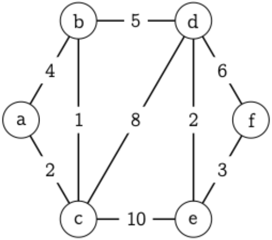
On appelle *matrice de pondération* d'un graphe $\mathscr{G}$ la matrice dont les coefficients correspondant aux sommets s et t valent :

$$A = \begin{cases} 0 & \text{si } t = s \\ \infty & \text{si } \{s, t\} \text{ n'est pas une arête} \\ p & \text{si } \{s, t\} \text{ est une arête de poids } p \end{cases}$$

Graphes pondérés

Exemple

$$\begin{pmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{pmatrix}$$



Sommaire

Coloriage d'un graphe

Rappels :

- Colorer un graphe, c'est affecter une couleur à chaque sommet de façon à ce que deux sommets adjacents soient de couleurs différentes

Coloriage d'un graphe

Rappels :

- Colorer un graphe, c'est affecter une couleur à chaque sommet de façon à ce que deux sommets adjacents soient de couleurs différentes
- Le nombre chromatique du graphe est le nombre minimal de couleurs nécessaire

Coloriage d'un graphe

Rappels :

- Colorer un graphe, c'est affecter une couleur à chaque sommet de façon à ce que deux sommets adjacents soient de couleurs différentes
- Le nombre chromatique du graphe est le nombre minimal de couleurs nécessaire

Exemple

Dans un groupe de TP de 14 étudiants, on doit former des groupes en faisant en sorte que les étudiants d'un même groupe ne s'entendent pas trop mal.

Modélisation

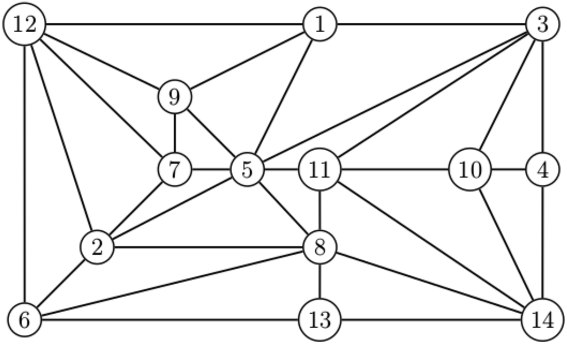
On représente la situation par un graphe simple non-orienté où :

- les sommets représentent les étudiants
- un arc entre deux sommets signifie que deux étudiants ne s'entendent pas

Modélisation

On représente la situation par un graphe simple non-orienté où :

- les sommets représentent les étudiants
- un arc entre deux sommets signifie que deux étudiants ne s'entendent pas



Algorithme Glouton

Entrées : liste ordonnée V des n sommets du graphe G ;
liste ordonnée C de couleurs

pour *i variant de 1 à n* **faire**

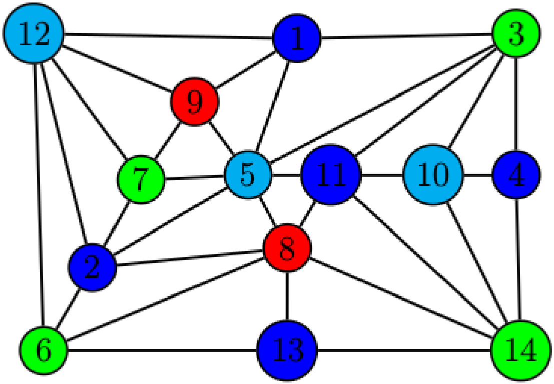
  $v = V[i]$;

  couleur = la première couleur de C non utilisée par les voisins de v ;

  colorer v en utilisant couleur ;

fin

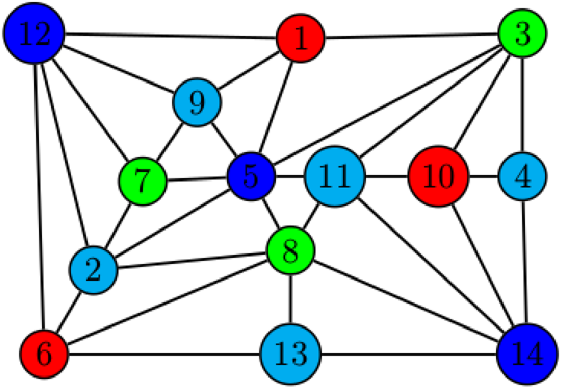
Algorithme Glouton



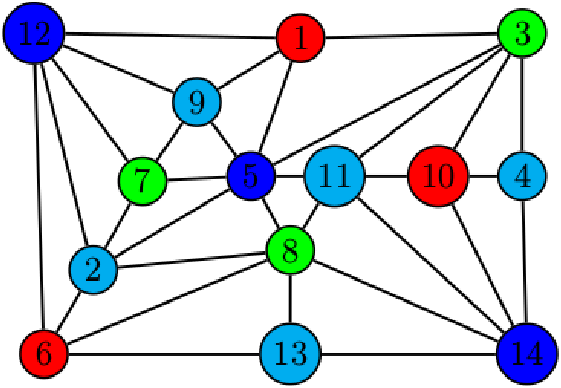
Algorithme de Welsch-Powell

Entrées : Graphe G avec ses sommets
Sorties : Les sommets affectés d'une couleur
début
    Trier les sommets de G par ordre de degré décroissant ;
    tant que *Tous les sommets de G ne sont pas colorés* **faire**
        Affecter à color une nouvelle couleur;
        Colorer avec color le 1^er sommet du tableau non encore coloré ;
        Colorer avec color tous les sommets de G non encore colorés et non adjacents à un sommet coloré avec color ;
    fin
    Afficher les sommets et leurs couleurs
fin

Algorithme de Welsch-Powell

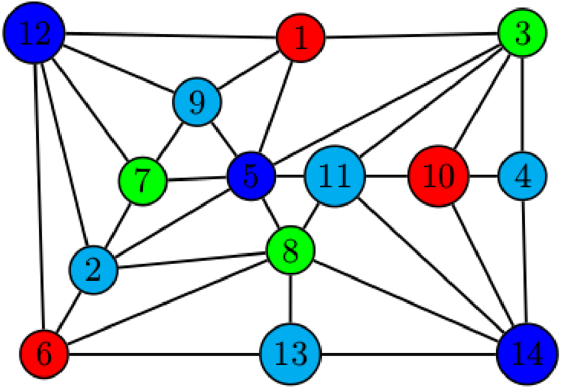


Algorithme de Welsch-Powell



- $s = 5$ de degré 8, en bleu avec s_{14} et s_{12}

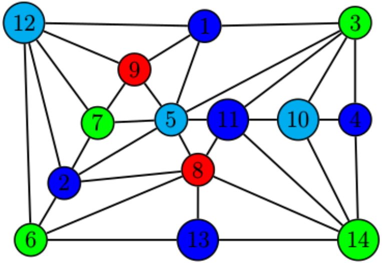
Algorithme de Welsch-Powell



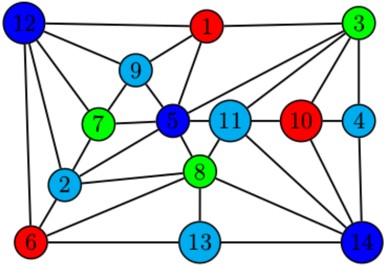
- $s = 5$ de degré 8, en bleu avec s_{14} et s_{12}
- $s = 8$ de degré 6, en vert avec s_3 et s_7, etc.

Coloration

Glouton :

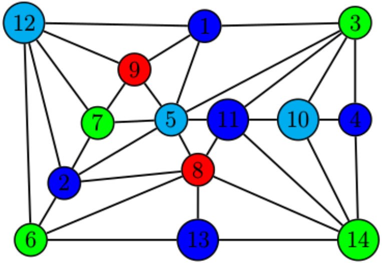


Welsch-Powell :

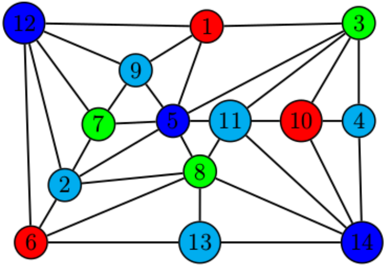


Coloration

Glouton :



Welsch-Powell :



Le résultat dépend de l'ordre de traitement des sommets

Coloration

Attention !

- les deux algorithmes ne donne pas toujours le nombre minimal de couleurs !

Coloration

Attention !

- les deux algorithmes ne donne pas toujours le nombre minimal de couleurs !
- L'algorithme Glouton peut être amélioré en traitant les sommets dans l'ordre décroissant de leur degré (comme dans l'algorithme de Welsh-Powell).

Attention !

- les deux algorithmes ne donne pas toujours le nombre minimal de couleurs !
- L'algorithme Glouton peut être amélioré en traitant les sommets dans l'ordre décroissant de leur degré (comme dans l'algorithme de Welsh-Powell).
- Le résultat d'un algorithme dépend fortement de l'ordre dans lequel les sommets sont traités

Parcours sur un graphe

- La première question que se pose un informaticien utilisant un graphe est de savoir comment parcourir les sommets

Parcours sur un graphe

- La première question que se pose un informaticien utilisant un graphe est de savoir comment parcourir les sommets
- De nombreux autres problèmes d'algorithmique se ramènent au parcours d'un graphe

Parcours sur un graphe

- La première question que se pose un informaticien utilisant un graphe est de savoir comment parcourir les sommets
- De nombreux autres problèmes d'algorithmique se ramènent au parcours d'un graphe
- Le résultat d'un parcours est un ensemble de chemins partants d'un sommet s allant vers les sommets accessibles depuis s.

Parcours sur un graphe

- La première question que se pose un informaticien utilisant un graphe est de savoir comment parcourir les sommets
- De nombreux autres problèmes d'algorithmique se ramènent au parcours d'un graphe
- Le résultat d'un parcours est un ensemble de chemins partants d'un sommet s allant vers les sommets accessibles depuis s.

Exemple

Le graphe peut représenter une arborescence de fichiers
Le problème peut-être une recherche de fichier

Parcours sur un graphe

- Le choix de l'ordre de visite de chaque sommet est primordial pour l'efficacité de l'algorithme

Parcours sur un graphe

- Le choix de l'ordre de visite de chaque sommet est primordial pour l'efficacité de l'algorithme
- Deux méthodes :

Parcours sur un graphe

- Le choix de l'ordre de visite de chaque sommet est primordial pour l'efficacité de l'algorithme
- Deux méthodes :
  - Parcours en largeur

Parcours sur un graphe

- Le choix de l'ordre de visite de chaque sommet est primordial pour l'efficacité de l'algorithme
- Deux méthodes :
  - Parcours en largeur
  - Parcours en profondeur

Parcours sur un graphe

- Le choix de l'ordre de visite de chaque sommet est primordial pour l'efficacité de l'algorithme
- Deux méthodes :
  - Parcours en largeur
  - Parcours en profondeur
- Dans un parcours un sommet peut avoir trois états :

Parcours sur un graphe

- Le choix de l'ordre de visite de chaque sommet est primordial pour l'efficacité de l'algorithme
- Deux méthodes :
  - Parcours en largeur
  - Parcours en profondeur
- Dans un parcours un sommet peut avoir trois états :
  - non visité ;

Parcours sur un graphe

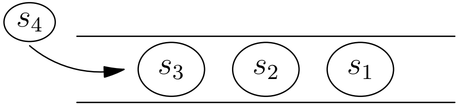
- Le choix de l'ordre de visite de chaque sommet est primordial pour l'efficacité de l'algorithme
- Deux méthodes :
  - Parcours en largeur
  - Parcours en profondeur
- Dans un parcours un sommet peut avoir trois états :
  - non visité ;
  - visité ;

Parcours sur un graphe

- Le choix de l'ordre de visite de chaque sommet est primordial pour l'efficacité de l'algorithme
- Deux méthodes :
  - Parcours en largeur
  - Parcours en profondeur
- Dans un parcours un sommet peut avoir trois états :
  - non visité ;
  - visité ;
  - ouvert lorsqu'il est en cours de traitement

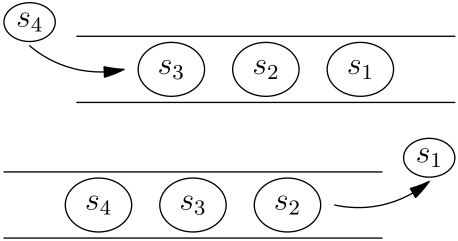
Parcours en largeur

Le *parcours en largeur* privilégie les sommets les plus proches en s'appuyant sur une file



Parcours en largeur

Le *parcours en largeur* privilégie les sommets les plus proches en s'appuyant sur une file



First in, first out

Parcours en largeur

Lorsque l'on fait un parcours en largeur à partir d'un sommet s,

- on atteint d'abord les voisins,

Parcours en largeur

Lorsque l'on fait un parcours en largeur à partir d'un sommet s,

- on atteint d'abord les voisins,
- ensuite les voisins des voisins (sauf ceux qui sont déjà atteints)

Parcours en largeur

Lorsque l'on fait un parcours en largeur à partir d'un sommet s,

- on atteint d'abord les voisins,
- ensuite les voisins des voisins (sauf ceux qui sont déjà atteints)
- et ainsi de suite.

Parcours en largeur

Lorsque l'on fait un parcours en largeur à partir d'un sommet s,

- on atteint d'abord les voisins,
- ensuite les voisins des voisins (sauf ceux qui sont déjà atteints)
- et ainsi de suite.

un sommet atteint sera toujours un sommet examiné

Parcours en largeur

Lorsque l'on fait un parcours en largeur à partir d'un sommet s,

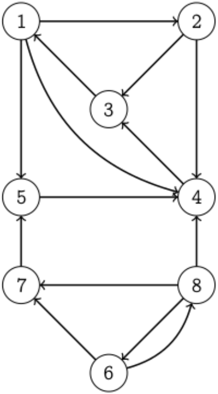
- on atteint d'abord les voisins,
- ensuite les voisins des voisins (sauf ceux qui sont déjà atteints)
- et ainsi de suite.

un sommet atteint sera toujours un sommet examiné

Voyons d'abord un exemple !

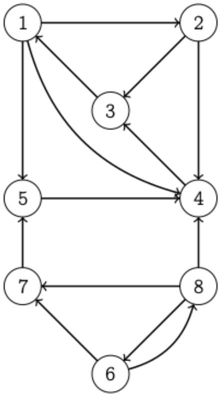
Parcours en largeur

- on commence par regarder 1 ;



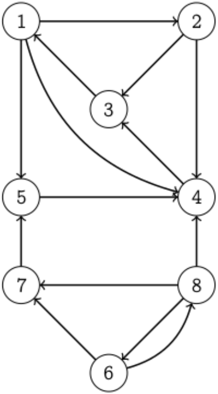
Parcours en largeur

- on commence par regarder 1 ;
- ses voisins 2, 4 et 5 sont placés dans la file ;
  file : 1-2-4-5



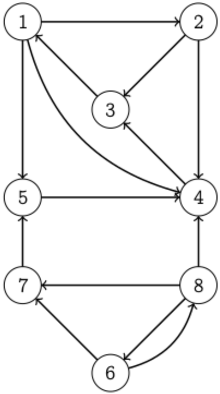
Parcours en largeur

- on commence par regarder 1 ;
- ses voisins 2, 4 et 5 sont placés dans la file ;
  file : 1-2-4-5
- 1 est traité puis marqué comme visité ;
  file : 2-4-5



Parcours en largeur

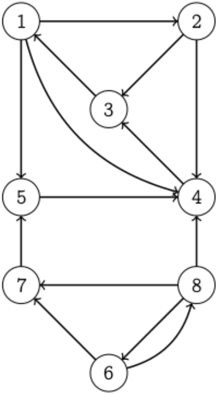
- on commence par regarder 1 ;
- ses voisins 2, 4 et 5 sont placés dans la file ;
  file : 1-2-4-5
- 1 est traité puis marqué comme visité ;
  file : 2-4-5



- On passe à 2 ; son voisin 3 est placé dans la file (4 est déjà dans la file)

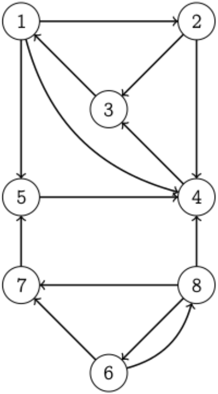
Parcours en largeur

- file : 2-4-5-3



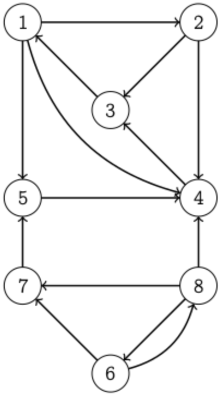
Parcours en largeur

- file : 2-4-5-3
- 2 est traité puis marqué comme visité ;
  file : 4-5-3



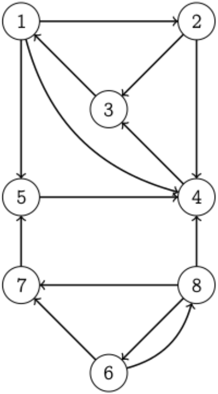
Parcours en largeur

- file : 2-4-5-3
- 2 est traité puis marqué comme visité ;
  file : 4-5-3
- le sommet suivant est 4. Son seul voisin est 3 qui a déjà été vu ;
  file : 5-3

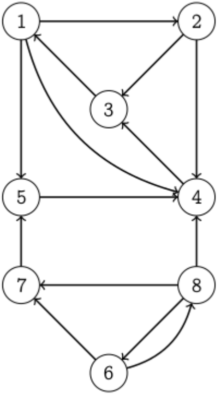


Parcours en largeur

- file : 2-4-5-3
- 2 est traité puis marqué comme visité ;
  file : 4-5-3
- le sommet suivant est 4. Son seul voisin est 3 qui a déjà été vu ;
  file : 5-3
- le sommet suivant est 5. Son seul voisin est 4, qui a déjà été vu ;
  file : 3



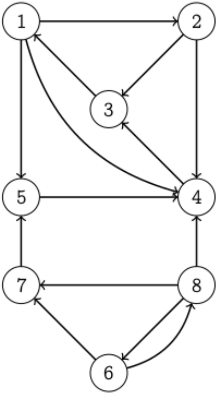
Parcours en largeur



- file : 2-4-5-3
- 2 est traité puis marqué comme visité ;
  file : 4-5-3
- le sommet suivant est 4. Son seul voisin est 3 qui a déjà été vu ;
  file : 5-3
- le sommet suivant est 5. Son seul voisin est 4, qui a déjà été vu ;
  file : 3
- le sommet suivant est 3. Le seul voisin est 1 qui a déjà été traité ;

Parcours en largeur

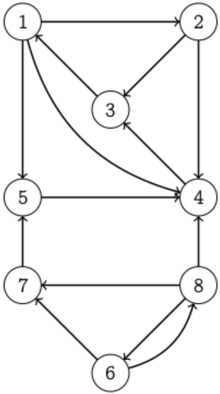
- file : 2-4-5-3
- 2 est traité puis marqué comme visité ;
  file : 4-5-3
- le sommet suivant est 4. Son seul voisin est 3 qui a déjà été vu ;
  file : 5-3
- le sommet suivant est 5. Son seul voisin est 4, qui a déjà été vu ;
  file : 3



- le sommet suivant est 3. Le seul voisin est 1 qui a déjà été traité ;
- la file est maintenant vide. On remonte donc maintenant à un sommet non encore atteint dans l'ordre des numéros croissant ;

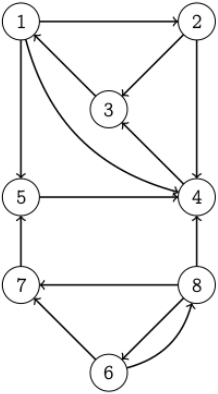
Parcours en largeur

- on regarde le sommet 6 dont les voisins sont 7 et 8 ;
  file : 6-7-8



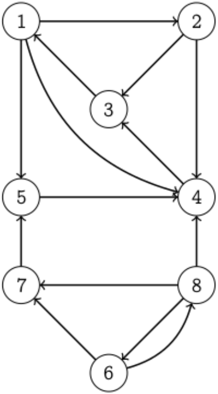
Parcours en largeur

- on regarde le sommet 6 dont les voisins sont 7 et 8 ;
  file : 6-7-8
- le sommet 6 est traité ;
  file : 7-8



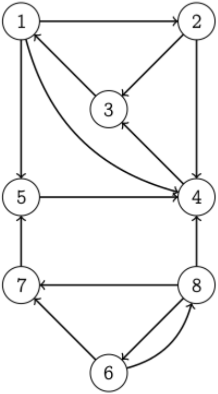
Parcours en largeur

- on regarde le sommet 6 dont les voisins sont 7 et 8 ;
  file : 6-7-8
- le sommet 6 est traité ;
  file : 7-8
- on traite 7 dont le seul voisin est 5 qui a déjà été traité ;
  file : 8



Parcours en largeur

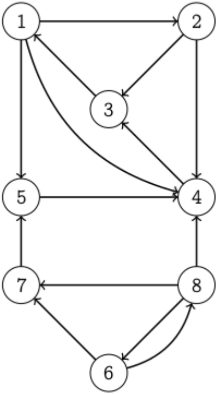
- on regarde le sommet 6 dont les voisins sont 7 et 8 ;
  file : 6-7-8
- le sommet 6 est traité ;
  file : 7-8
- on traite 7 dont le seul voisin est 5 qui a déjà été traité ;
  file : 8



- on traite 8 dont les voisins 4, 6 et 7 ont tous été déjà traités ;

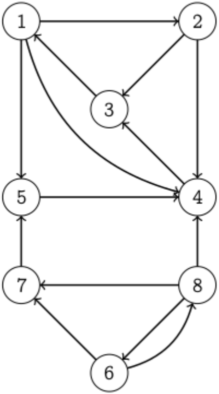
Parcours en largeur

- on regarde le sommet 6 dont les voisins sont 7 et 8 ;
  file : 6-7-8
- le sommet 6 est traité ;
  file : 7-8
- on traite 7 dont le seul voisin est 5 qui a déjà été traité ;
  file : 8



- on traite 8 dont les voisins 4, 6 et 7 ont tous été déjà traités ;
- la file est maintenant vide et tous les sommets ont été traités.

Parcours en largeur



Le parcours en largeur du graphe est donc

$$1 - 2 - 4 - 5 - 3 - 6 - 7 - 8$$

Parcours en largeur

Principe de l'algorithme

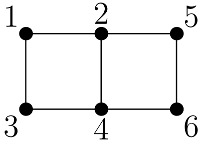
début

1. Mettre le sommet source dans la file ;
2. Mettre tous les sommets voisins non explorés dans la file ;
3. Retirer le sommet du début de la file pour l'examiner ;
4. Si la file n'est pas vide reprendre à l'étape 2 ;

fin

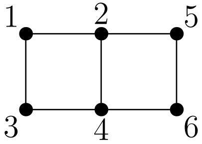
Parcours en largeur

Déterminer le parcours en largeur du graphe :



Parcours en largeur

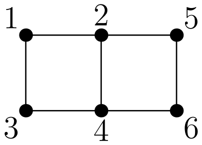
Déterminer le parcours en largeur du graphe :



- on commence avec le sommet 1 ;

Parcours en largeur

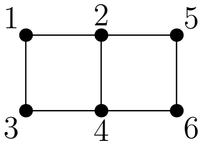
Déterminer le parcours en largeur du graphe :



- on commence avec le sommet 1 ;
- les sommets 2 et 3 sont placés dans la file ;

Parcours en largeur

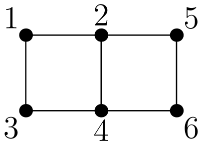
Déterminer le parcours en largeur du graphe :



- on commence avec le sommet 1 ;
- les sommets 2 et 3 sont placés dans la file ;
- le sommet 1 est examiné puis marqué ;

Parcours en largeur

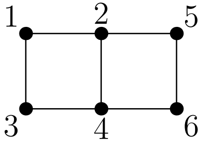
Déterminer le parcours en largeur du graphe :



- on commence avec le sommet 1 ;
- les sommets 2 et 3 sont placés dans la file ;
- le sommet 1 est examiné puis marqué ;
- le sommet 2 est le suivant dans la file ;

Parcours en largeur

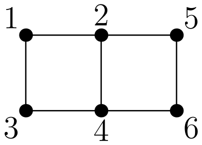
Déterminer le parcours en largeur du graphe :



- on commence avec le sommet 1 ;
- les sommets 2 et 3 sont placés dans la file ;
- le sommet 1 est examiné puis marqué ;
- le sommet 2 est le suivant dans la file ;
- ses voisins 4 et 5 sont mis dans la file ;

Parcours en largeur

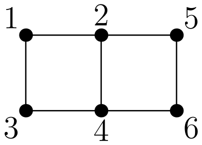
Déterminer le parcours en largeur du graphe :



- on commence avec le sommet 1 ;
- les sommets 2 et 3 sont placés dans la file ;
- le sommet 1 est examiné puis marqué ;
- le sommet 2 est le suivant dans la file ;
- ses voisins 4 et 5 sont mis dans la file ;
- le sommet 2 est examiné puis marqué ;

Parcours en largeur

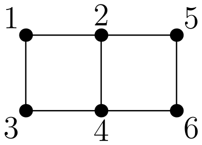
Déterminer le parcours en largeur du graphe :



- on commence avec le sommet 1 ;
- les sommets 2 et 3 sont placés dans la file ;
- le sommet 1 est examiné puis marqué ;
- le sommet 2 est le suivant dans la file ;
- ses voisins 4 et 5 sont mis dans la file ;
- le sommet 2 est examiné puis marqué ;
- le sommet 3 est le suivant ; il n'a pas de voisin non marqué

Parcours en largeur

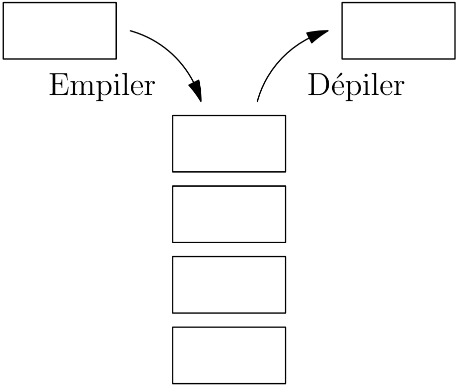
Déterminer le parcours en largeur du graphe :



- on commence avec le sommet 1 ;
- les sommets 2 et 3 sont placés dans la file ;
- le sommet 1 est examiné puis marqué ;
- le sommet 2 est le suivant dans la file ;
- ses voisins 4 et 5 sont mis dans la file ;
- le sommet 2 est examiné puis marqué ;
- le sommet 3 est le suivant ; il n'a pas de voisin non marqué
- les sommets 4, 5 et 6 sont examinés...

Parcours en profondeur

Le parcours en profondeur privilégie les sommets les plus éloignés en s'appuyant sur une pile



Last in, First out

Parcours en profondeur

lorsque l'on fait un parcours en profondeur à partir d'un sommet s,

- on tente d'avancer le plus loin possible dans le graphe

Parcours en profondeur

lorsque l'on fait un parcours en profondeur à partir d'un sommet s,

- on tente d'avancer le plus loin possible dans le graphe
- lorsque toutes les possibilités de progression sont bloquées, on revient pour explorer un nouveau chemin ou une nouvelle chaîne.

Parcours en profondeur

lorsque l'on fait un parcours en profondeur à partir d'un sommet s,

- on tente d'avancer le plus loin possible dans le graphe
- lorsque toutes les possibilités de progression sont bloquées, on revient pour explorer un nouveau chemin ou une nouvelle chaîne.
- Concrètement,
  - on part d'un sommet, on va voir un de ses voisins puis un voisin du voisin, ...

Parcours en profondeur

lorsque l'on fait un parcours en profondeur à partir d'un sommet s,

- on tente d'avancer le plus loin possible dans le graphe
- lorsque toutes les possibilités de progression sont bloquées, on revient pour explorer un nouveau chemin ou une nouvelle chaîne.
- Concrètement,
  - on part d'un sommet, on va voir un de ses voisins puis un voisin du voisin, ...
  - lorsque l'on est bloqué, alors on va en arrière.

Parcours en profondeur

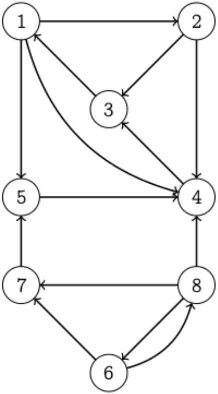
lorsque l'on fait un parcours en profondeur à partir d'un sommet s,

- on tente d'avancer le plus loin possible dans le graphe
- lorsque toutes les possibilités de progression sont bloquées, on revient pour explorer un nouveau chemin ou une nouvelle chaîne.
- Concrètement,
  - on part d'un sommet, on va voir un de ses voisins puis un voisin du voisin, ...
  - lorsque l'on est bloqué, alors on va en arrière.

Regardons un exemple

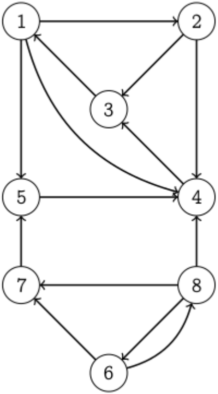
Parcours en profondeur

- on commence par 1 ;



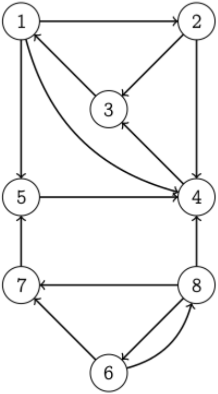
Parcours en profondeur

- on commence par 1 ;
- le sommet 1 est traité puis marqué comme visité ;



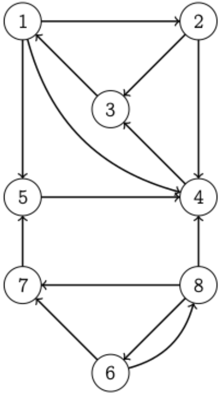
Parcours en profondeur

- on commence par 1 ;
- le sommet 1 est traité puis marqué comme visité ;
- les voisins 2, 4 et 5 sont placés dans la pile ;
  pile : 2-4-5



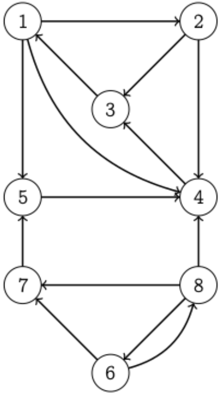
Parcours en profondeur

- on commence par 1 ;
- le sommet 1 est traité puis marqué comme visité ;
- les voisins 2, 4 et 5 sont placés dans la pile ;
  pile : 2-4-5
- le sommet 5 est traité puis marqué ;



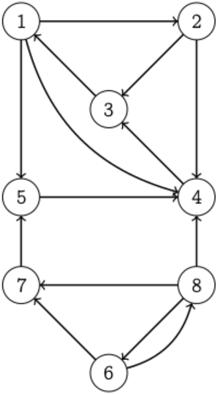
Parcours en profondeur

- on commence par 1 ;
- le sommet 1 est traité puis marqué comme visité ;
- les voisins 2, 4 et 5 sont placés dans la pile ;
  pile : 2-4-5
- le sommet 5 est traité puis marqué ;
  - son voisin 4 est déjà dans la pile
    pile : 2-4



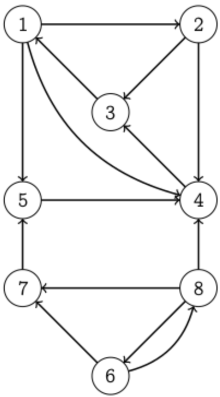
Parcours en profondeur

- pile : 2-4



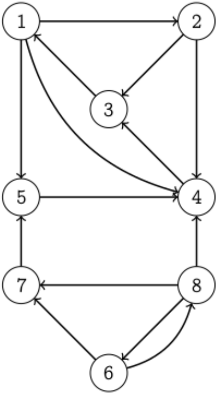
Parcours en profondeur

- pile : 2-4
- le sommet 4 est traité puis marqué ;



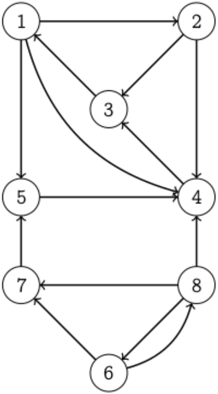
Parcours en profondeur

- pile : 2-4
- le sommet 4 est traité puis marqué ;
- son seul voisin 3 est placé dans la pile ;
  pile : 2-3



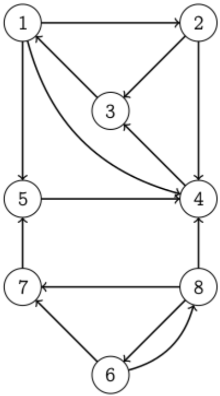
Parcours en profondeur

- pile : 2-4
- le sommet 4 est traité puis marqué ;
- son seul voisin 3 est placé dans la pile ;
  pile : 2-3
- le sommet 3 est traité puis marqué ;



Parcours en profondeur

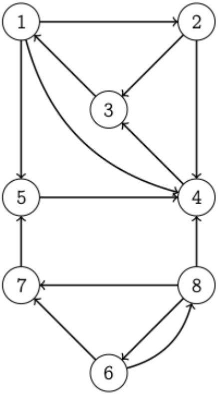
- pile : 2-4
- le sommet 4 est traité puis marqué ;
- son seul voisin 3 est placé dans la pile ;
  pile : 2-3
- le sommet 3 est traité puis marqué ;
- on est bloqué : le sommet 3 a pour seul voisin 1 qui a déjà été traité ;
  pile : 2



Parcours en profondeur

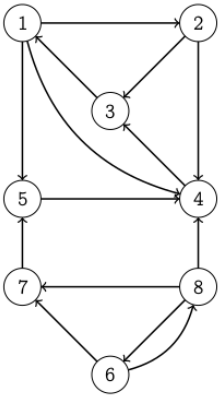
- pile : 2-4
- le sommet 4 est traité puis marqué ;
- son seul voisin 3 est placé dans la pile ;
  pile : 2-3
- le sommet 3 est traité puis marqué ;
- on est bloqué : le sommet 3 a pour seul voisin 1 qui a déjà été traité ;
  pile : 2

  - on examine 2 ;



Parcours en profondeur

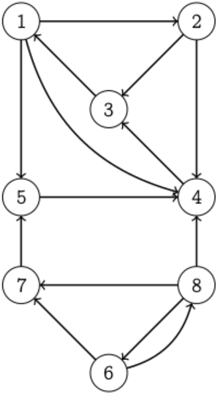
- pile : 2-4
- le sommet 4 est traité puis marqué ;
- son seul voisin 3 est placé dans la pile ;
  pile : 2-3
- le sommet 3 est traité puis marqué ;
- on est bloqué : le sommet 3 a pour seul voisin 1 qui a déjà été traité ;
  pile : 2



- on examine 2 ;
- on est bloqué : les voisins de 2 sont 3 et 4 et ont déjà été traités ;

Parcours en profondeur

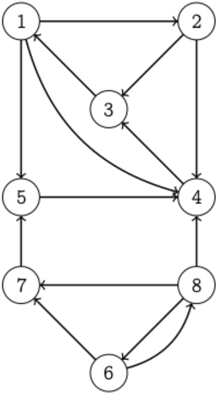
- pile : 2-4
- le sommet 4 est traité puis marqué ;
- son seul voisin 3 est placé dans la pile ;
  pile : 2-3
- le sommet 3 est traité puis marqué ;
- on est bloqué : le sommet 3 a pour seul voisin 1 qui a déjà été traité ;
  pile : 2



- on examine 2 ;
- on est bloqué : les voisins de 2 sont 3 et 4 et ont déjà été traités ;
- La pile est vide ; on passe au sommet 6

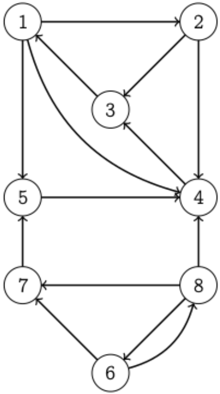
Parcours en profondeur

- 6 est traité puis marqué ;



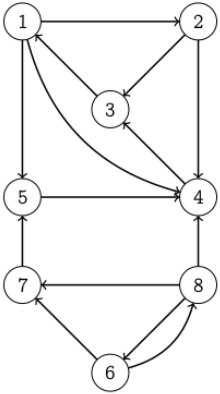
Parcours en profondeur

- 6 est traité puis marqué ;
- Les sommets 7 et 8, voisins de 6 sont placés dans la pile ;
  pile : 7-8



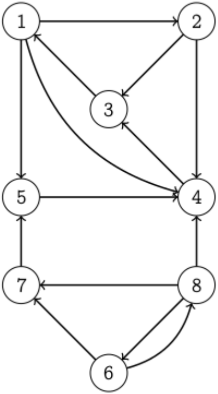
Parcours en profondeur

- 6 est traité puis marqué ;
- Les sommets 7 et 8, voisins de 6 sont placés dans la pile ;
  pile : 7-8
- On visite 8 puis 7



Parcours en profondeur

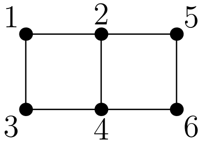
- 6 est traité puis marqué ;
- Les sommets 7 et 8, voisins de 6 sont placés dans la pile ;
  pile : 7-8
- On visite 8 puis 7



Parcours en profondeur : 1-5-4-3-2-6-8-7

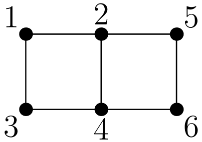
Parcours en profondeur

Déterminer le parcours en profondeur du graphe :



Parcours en profondeur

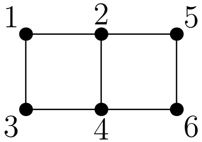
Déterminer le parcours en profondeur du graphe :



- on commence par le sommet 1 ;

Parcours en profondeur

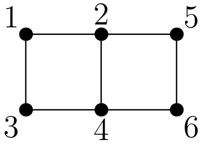
Déterminer le parcours en profondeur du graphe :



- on commence par le sommet 1 ;
- 1 est traité puis marqué ;

Parcours en profondeur

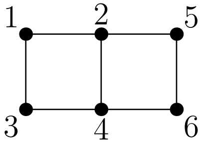
Déterminer le parcours en profondeur du graphe :



- on commence par le sommet 1 ;
- 1 est traité puis marqué ;
- les sommets 2 et 3 sont placés dans la pile ;

Parcours en profondeur

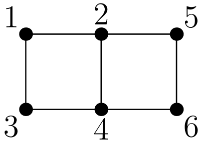
Déterminer le parcours en profondeur du graphe :



- on commence par le sommet 1 ;
- 1 est traité puis marqué ;
- les sommets 2 et 3 sont placés dans la pile ;
- 3 est le suivant ; son voisin 4 est mis dans la pile ;

Parcours en profondeur

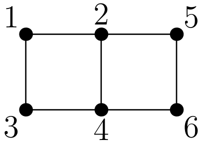
Déterminer le parcours en profondeur du graphe :



- on commence par le sommet 1 ;
- 1 est traité puis marqué ;
- les sommets 2 et 3 sont placés dans la pile ;
- 3 est le suivant ; son voisin 4 est mis dans la pile ;
- 3 est traité puis marqué ;

Parcours en profondeur

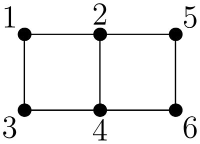
Déterminer le parcours en profondeur du graphe :



- on commence par le sommet 1 ;
- 1 est traité puis marqué ;
- les sommets 2 et 3 sont placés dans la pile ;
- 3 est le suivant ; son voisin 4 est mis dans la pile ;
- 3 est traité puis marqué ;
- 4 est le suivant ; son voisin 6 est mis dans la pile ;

Parcours en profondeur

Déterminer le parcours en profondeur du graphe :



- on commence par le sommet 1 ;
- 1 est traité puis marqué ;
- les sommets 2 et 3 sont placés dans la pile ;
- 3 est le suivant ; son voisin 4 est mis dans la pile ;
- 3 est traité puis marqué ;
- 4 est le suivant ; son voisin 6 est mis dans la pile ;
- On réitère et on obtient 1-3-4-6-5-2