

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

Généralités

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

Définitions

- Un graphe $G = (V, E)$ est un couple constitué d'un ensemble :

Définitions

- Un graphe $G = (V, E)$ est un couple constitué d'un ensemble :
 - V d'éléments appelés *sommets* (*vertices*);

Définitions

- Un graphe $G = (V, E)$ est un couple constitué d'un ensemble :
 - V d'éléments appelés *sommets* (*vertices*);
 - $E \subset V \times V$ d'éléments appelés *arcs* (*edges*).

Définitions

- Un graphe $G = (V, E)$ est un couple constitué d'un ensemble :
 - V d'éléments appelés *sommets* (*vertices*);
 - $E \subset V \times V$ d'éléments appelés *arcs* (*edges*).
- $(x, x) \in E$ est une *boucle*

Définitions

- Un graphe $G = (V, E)$ est un couple constitué d'un ensemble :
 - V d'éléments appelés *sommets* (*vertices*);
 - $E \subset V \times V$ d'éléments appelés *arcs* (*edges*).
- $(x, x) \in E$ est une *boucle*
- Si E est symétrique : graphe *non orienté* (GNO) et *arête* (notée $\{x, y\}$) plutôt qu'arc. Sinon c'est un graphe *orienté* (GO).

Définitions

- Un graphe $G = (V, E)$ est un couple constitué d'un ensemble :
 - V d'éléments appelés *sommets* (*vertices*);
 - $E \subset V \times V$ d'éléments appelés *arcs* (*edges*).
- $(x, x) \in E$ est une *boucle*
- Si E est symétrique : graphe *non orienté* (GNO) et *arête* (notée $\{x, y\}$) plutôt qu'arc. Sinon c'est un graphe *orienté* (GO).
- $\text{card}(V) = |V|$: *ordre* du graphe (parfois noté $|G|$)

Définitions

- Un graphe $G = (V, E)$ est un couple constitué d'un ensemble :
 - V d'éléments appelés *sommets* (*vertices*);
 - $E \subset V \times V$ d'éléments appelés *arcs* (*edges*).
- $(x, x) \in E$ est une *boucle*
- Si E est symétrique : graphe *non orienté* (GNO) et *arête* (notée $\{x, y\}$) plutôt qu'arc. Sinon c'est un graphe *orienté* (GO).
- $\text{card}(V) = |V|$: *ordre* du graphe (parfois noté $|G|$)
- Si $xy \in E$: x et y sont *adjacents* (ou *voisins*)

Définitions

- Un graphe $G = (V, E)$ est un couple constitué d'un ensemble :
 - V d'éléments appelés *sommets* (*vertices*);
 - $E \subset V \times V$ d'éléments appelés *arcs* (*edges*).
- $(x, x) \in E$ est une *boucle*
- Si E est symétrique : graphe *non orienté* (GNO) et *arête* (notée $\{x, y\}$) plutôt qu'arc. Sinon c'est un graphe *orienté* (GO).
- $\text{card}(V) = |V|$: *ordre* du graphe (parfois noté $|G|$)
- Si $xy \in E$: x et y sont *adjacents* (ou *voisins*)
- $N(x)$: ensemble des voisins (*neighbours*) de x

Définitions

- G *simple* : au plus une arête reliant deux sommets (sinon *multigraphe*) et pas de boucle

Définitions

- G *simple* : au plus une arête reliant deux sommets (sinon *multigraphe*) et pas de boucle
- G *complet* : simple et tous les sommets sont adjacents

Définitions

- G *simple* : au plus une arête reliant deux sommets (sinon *multigraphe*) et pas de boucle
- G *complet* : simple et tous les sommets sont adjacents
- *degré* d'un sommet x (noté $d(x)$) : $|N(x)|$

Définitions

- G *simple* : au plus une arête reliant deux sommets (sinon *multigraphe*) et pas de boucle
- G *complet* : simple et tous les sommets sont adjacents
- *degré* d'un sommet x (noté $d(x)$) : $|N(x)|$
- graphe k -*régulier* : $\forall x \in V \quad d(x) = k$

Définitions

- G *simple* : au plus une arête reliant deux sommets (sinon *multigraphe*) et pas de boucle
- G *complet* : simple et tous les sommets sont adjacents
- *degré* d'un sommet x (noté $d(x)$) : $|N(x)|$
- graphe k -*régulier* : $\forall x \in V \quad d(x) = k$
- Pour un GO, on distingue :

Définitions

- G *simple* : au plus une arête reliant deux sommets (sinon *multigraphe*) et pas de boucle
- G *complet* : simple et tous les sommets sont adjacents
- *degré* d'un sommet x (noté $d(x)$) : $|N(x)|$
- graphe k -*régulier* : $\forall x \in V \quad d(x) = k$
- Pour un GO, on distingue :
 - *degré sortant* ($d^+(x)$) : nombre d'arcs ayant x pour origine ;

Définitions

- G *simple* : au plus une arête reliant deux sommets (sinon *multigraphe*) et pas de boucle
- G *complet* : simple et tous les sommets sont adjacents
- *degré* d'un sommet x (noté $d(x)$) : $|N(x)|$
- graphe k -*régulier* : $\forall x \in V \quad d(x) = k$
- Pour un GO, on distingue :
 - *degré sortant* ($d^+(x)$) : nombre d'arcs ayant x pour origine ;
 - *degré entrant* ($d^-(x)$) : nombre d'arcs ayant x pour extrémité ;

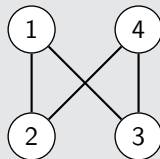
Exemple

Graphe non orienté d'ordre 4

$V = \{1, 2, 3, 4\}$

$E = \{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{3, 4\}\}$

simple, non complet, 2-régulier



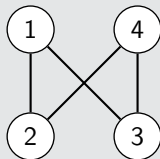
Exemple

Graphe non orienté d'ordre 4

$$V = \{1, 2, 3, 4\}$$

$$E = \{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{3, 4\}\}$$

simple, non complet, 2-régulier



Exemple

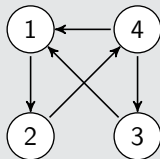
Graphe orienté d'ordre 4

$$V = \{1, 2, 3, 4\}$$

$$E = \{(1, 2), (3, 1), (2, 4), (4, 3), (4, 1)\}$$

simple, non complet

$$d^-(1) = 2; d^+(3) = 1$$



Question

Existe-t-il un graphe d'ordre 3 dont les sommets ont pour degrés respectifs 1, 2, 2 ?

Question

Existe-t-il un graphe d'ordre 3 dont les sommets ont pour degrés respectifs 1, 2, 2 ?

Proposition

Soit $G = (V, E)$ un GNO.

$$\sum_{x \in V} d(x) = 2|E|$$

Soit $G = (V, E)$ un GO.

$$\sum_{x \in V} (d^+(x) + d^-(x)) = 2|E|$$

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

Matrice d'adjacence

Définition

Soit $G = (V, E)$ un graphe. La *matrice d'adjacence* de G est définie par :

$$M_{ij} = \begin{cases} 1 & \text{si } ij \in E \\ 0 & \text{sinon} \end{cases}$$

Pour un GNO, M est symétrique.

Matrice d'adjacence

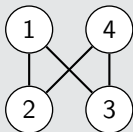
Définition

Soit $G = (V, E)$ un graphe. La *matrice d'adjacence* de G est définie par :

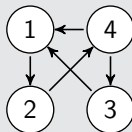
$$M_{ij} = \begin{cases} 1 & \text{si } ij \in E \\ 0 & \text{sinon} \end{cases}$$

Pour un GNO, M est symétrique.

Exemple



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

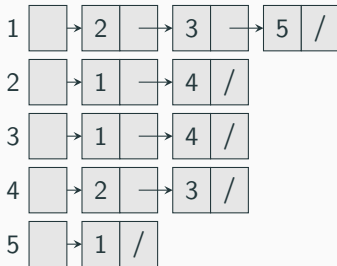
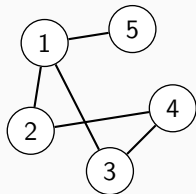


$$M = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

Liste d'adjacence

Définition

On mémorise pour chaque sommet la liste de ses voisins.



Structure de liste (doublement) chaînée.

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

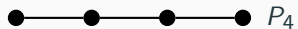
Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

- chaîne P_n (*path*)

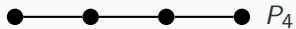


- chaîne P_n (*path*)

- cycle C_n



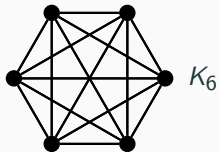
- chaîne P_n (*path*)



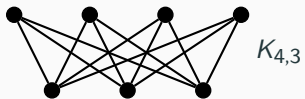
- cycle C_n



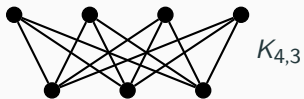
- complet K_n



- biparti complet $K_{n,p}$



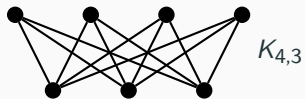
- biparti complet $K_{n,p}$



- roue W_n (*wheel*)



- biparti complet $K_{n,p}$



- roue W_n (*wheel*)



- étoile S_n (en fait c'est $K_{1,n-1}$!)



1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

Définition

Un *sous-graphe* de $G = (V, E)$ est de la forme $G' = (V', E')$ avec $V' \subset V$ et $E' \subset E$.

Définition

Un *sous-graphe* de $G = (V, E)$ est de la forme $G' = (V', E')$ avec $V' \subset V$ et $E' \subset E$.

Définition

Un *sous-graphe induit* de G est de la forme $G' = (V', E')$ avec $V' \subset V$, et $E' = \{xy \in E \mid x \in V' \wedge y \in V'\}$. Autrement, on retire des sommets, et on ne garde que les arêtes qui existent encore.

Un sous-graphe complet est appelé une *clique*.

Définition

Un *sous-graphe* de $G = (V, E)$ est de la forme $G' = (V', E')$ avec $V' \subset V$ et $E' \subset E$.

Définition

Un *sous-graphe induit* de G est de la forme $G' = (V', E')$ avec $V' \subset V$, et $E' = \{xy \in E \mid x \in V' \wedge y \in V'\}$. Autrement, on retire des sommets, et on ne garde que les arêtes qui existent encore.

Un sous-graphe complet est appelé une *clique*.

Définition

Un *sous-graphe partiel* de G est de la forme $G' = (V, E')$ avec $E' \subset E$. Autrement, on garde tous les sommets, et on retire des arêtes.

Exemple

K_4 est un sous-graphe induit de K_5 .

Exemple

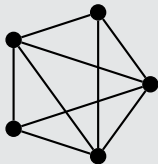
K_4 est un sous-graphe induit de K_5 .

P_4 est un sous-graphe partiel de C_4 .

Exemple

K_4 est un sous-graphe induit de K_5 .

P_4 est un sous-graphe partiel de C_4 .

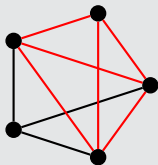


Ce graphe contient deux 4-cliques.

Exemple

K_4 est un sous-graphe induit de K_5 .

P_4 est un sous-graphe partiel de C_4 .

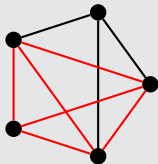


Ce graphe contient deux 4-cliques.

Exemple

K_4 est un sous-graphe induit de K_5 .

P_4 est un sous-graphe partiel de C_4 .



Ce graphe contient deux 4-cliques.

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

Définition

Deux graphes $G = (V, E)$ et $G' = (V', E')$ sont *isomorphes* s'il existe une bijection $\varphi : V \rightarrow V'$ telle que

$$\forall x, y \in V \quad xy \in E \iff \varphi(x)\varphi(y) \in E' .$$

Si $G = G'$, on dit que φ est un *automorphisme*.

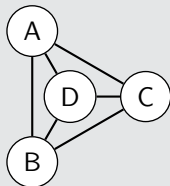
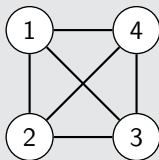
Définition

Deux graphes $G = (V, E)$ et $G' = (V', E')$ sont *isomorphes* s'il existe une bijection $\varphi : V \rightarrow V'$ telle que

$$\forall x, y \in V \quad xy \in E \iff \varphi(x)\varphi(y) \in E'.$$

Si $G = G'$, on dit que φ est un *automorphisme*.

Exemple



$$\varphi : \{1, 2, 3, 4\} \rightarrow \{A, B, C, D\}$$

$$\varphi(1) = A$$

$$\varphi(2) = B$$

$$\varphi(3) = C$$

$$\varphi(4) = D$$

Définition

Un graphe est *planaire* s'il admet une représentation sans croisement d'arêtes.

Définition

Un graphe est *planaire* s'il admet une représentation sans croisement d'arêtes.

Exemple

D'après l'exemple précédent, K_4 est planaire.

Pour prouver qu'un graphe est planaire, il suffit de le représenter correctement. Mais pour prouver qu'il ne l'est pas ...

Jeu : planarity.net

Androïd : Untangle et bien d'autres

Graphes et chemins

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

Définitions

- *chaîne* (resp. *cycle*) d'un GNO G : sous-graphe de G qui est lui-même une chaîne (resp. un cycle)

Définitions

- *chaîne* (resp. *cycle*) d'un GNO G : sous-graphe de G qui est lui-même une chaîne (resp. un cycle)
- *longueur* d'une chaîne (resp. cycle) : nombre d'arêtes qui la composent

Définitions

- *chaîne* (resp. *cycle*) d'un GNO G : sous-graphe de G qui est lui-même une chaîne (resp. un cycle)
- *longueur* d'une chaîne (resp. cycle) : nombre d'arêtes qui la composent
- Pour un GO, chaîne \rightarrow *chemin* et cycle \rightarrow *circuit*.

Définitions

- *chaîne* (resp. *cycle*) d'un GNO G : sous-graphe de G qui est lui-même une chaîne (resp. un cycle)
- *longueur* d'une chaîne (resp. cycle) : nombre d'arêtes qui la composent
- Pour un GO, chaîne \rightarrow *chemin* et cycle \rightarrow *circuit*.

Remarque

- Ces définitions correspondent à la notion de chaînes et cycles *élémentaires* chez certains auteurs : les sommets (et donc les arêtes) empruntés sont distincts.
- Lorsque l'on emprunte plusieurs fois la même arête et/ou le même sommet, on parlera de parcours ou de marche.

Proposition

Tout GNO sans cycle avec $|E| \geq 1$ possède au moins deux sommets pendants (de degré 1).

Proposition

Tout GNO sans cycle avec $|E| \geq 1$ possède au moins deux sommets pendants (de degré 1).

Démonstration.

Considérons une chaîne de longueur maximale (x_1, x_2, \dots, x_p) . $p \geq 2$



Proposition

Tout GNO sans cycle avec $|E| \geq 1$ possède au moins deux sommets pendants (de degré 1).

Démonstration.

Considérons une chaîne de longueur maximale (x_1, x_2, \dots, x_p) . $p \geq 2$
Si $d(x_1) \geq 2$, il admet un autre voisin $z \neq x_2$.



Proposition

Tout GNO sans cycle avec $|E| \geq 1$ possède au moins deux sommets pendants (de degré 1).

Démonstration.

Considérons une chaîne de longueur maximale (x_1, x_2, \dots, x_p) . $p \geq 2$

Si $d(x_1) \geq 2$, il admet un autre voisin $z \neq x_2$.

Si $p \geq 3$, $z \notin \{x_3, \dots, x_p\}$, car il y aurait un cycle, ce qui est exclu. Donc z n'est pas dans la chaîne maximale.



Proposition

Tout GNO sans cycle avec $|E| \geq 1$ possède au moins deux sommets pendants (de degré 1).

Démonstration.

Considérons une chaîne de longueur maximale (x_1, x_2, \dots, x_p) . $p \geq 2$

Si $d(x_1) \geq 2$, il admet un autre voisin $z \neq x_2$.

Si $p \geq 3$, $z \notin \{x_3, \dots, x_p\}$, car il y aurait un cycle, ce qui est exclu. Donc z n'est pas dans la chaîne maximale.

Mais alors $(z, x_1, x_2, \dots, x_p)$ est une chaîne de longueur supérieure !



Proposition

Tout GNO sans cycle avec $|E| \geq 1$ possède au moins deux sommets pendants (de degré 1).

Démonstration.

Considérons une chaîne de longueur maximale (x_1, x_2, \dots, x_p) . $p \geq 2$

Si $d(x_1) \geq 2$, il admet un autre voisin $z \neq x_2$.

Si $p \geq 3$, $z \notin \{x_3, \dots, x_p\}$, car il y aurait un cycle, ce qui est exclu. Donc z n'est pas dans la chaîne maximale.

Mais alors $(z, x_1, x_2, \dots, x_p)$ est une chaîne de longueur supérieure !

De même $d(x_p) = 1$. □

Soit M la matrice d'adjacence d'un graphe, et soit $p \in \mathbb{N}^*$.

Soit M la matrice d'adjacence d'un graphe, et soit $p \in \mathbb{N}^*$.

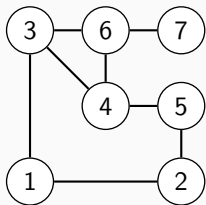
Proposition

Le coefficient $(M^p)_{i,j}$ est le nombre de parcours de longueur p reliant les sommets i et j .

Soit M la matrice d'adjacence d'un graphe, et soit $p \in \mathbb{N}^*$.

Proposition

Le coefficient $(M^p)_{i,j}$ est le nombre de parcours de longueur p reliant les sommets i et j .

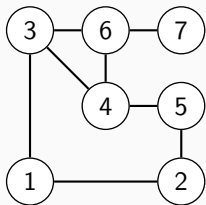


$$M^3 = \begin{pmatrix} 0 & 3 & 4 & 2 & 1 & 1 & 1 \\ 3 & 0 & 1 & 1 & 3 & 2 & 0 \\ 4 & 1 & 2 & 5 & 2 & 5 & 1 \\ 2 & 1 & 5 & 2 & 4 & 5 & 1 \\ 1 & 3 & 2 & 4 & 0 & 1 & 1 \\ 1 & 2 & 5 & 5 & 1 & 2 & 3 \\ 1 & 0 & 1 & 1 & 1 & 3 & 0 \end{pmatrix}$$

Soit M la matrice d'adjacence d'un graphe, et soit $p \in \mathbb{N}^*$.

Proposition

Le coefficient $(M^p)_{i,j}$ est le nombre de parcours de longueur p reliant les sommets i et j .



$$M^3 = \begin{pmatrix} 0 & 3 & 4 & 2 & 1 & 1 & 1 \\ 3 & 0 & 1 & 1 & 3 & 2 & 0 \\ 4 & 1 & 2 & 5 & 2 & 5 & 1 \\ 2 & 1 & 5 & 2 & 4 & 5 & 1 \\ 1 & 3 & 2 & 4 & 0 & 1 & 1 \\ 1 & 2 & 5 & 5 & 1 & 2 & 3 \\ 1 & 0 & 1 & 1 & 1 & 3 & 0 \end{pmatrix}$$

Il y a 5 parcours de longueur 3 reliant le sommet 3 au sommet 4.
Lesquels ?

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

Définitions

- Un GNO est dit *connexe* si, pour toute paire $\{x,y\} \subset V$, il existe une chaîne reliant x à y .

Définitions

- Un GNO est dit *connexe* si, pour toute paire $\{x, y\} \subset V$, il existe une chaîne reliant x à y .
- UN GO est dit *fortement connexe* si, pour tout couple $(x, y) \in V^2$, il existe un chemin d'origine x et d'extrémité y .

Définitions

- Un GNO est dit *connexe* si, pour toute paire $\{x, y\} \subset V$, il existe une chaîne reliant x à y .
- UN GO est dit *fortement connexe* si, pour tout couple $(x, y) \in V^2$, il existe un chemin d'origine x et d'extrémité y .

Relation

Soit G un graphe, non nécessairement connexe. Pour $x, y \in V$ on définit :

$$x \mathcal{R} y \Leftrightarrow \text{il existe une chaîne reliant } x \text{ à } y$$

\mathcal{R} est une relation d'équivalence ; les classes d'équivalence sont appelées *composantes connexes* de G .

Un GNO est donc connexe s'il n'a qu'une seule composante connexe.

Soit G un GNO connexe.

Définitions

- La *distance* entre deux sommets est la longueur minimale d'une chaîne reliant x à y . On la note $d(x,y)$.

Soit G un GNO connexe.

Définitions

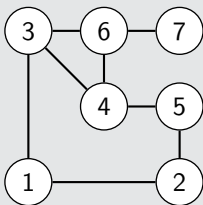
- La *distance* entre deux sommets est la longueur minimale d'une chaîne reliant x à y . On la note $d(x,y)$.
- Le *diamètre* de G est la plus grande distance séparant deux de ses sommets.

Soit G un GNO connexe.

Définitions

- La *distance* entre deux sommets est la longueur minimale d'une chaîne reliant x à y . On la note $d(x,y)$.
- Le *diamètre* de G est la plus grande distance séparant deux de ses sommets.

Exemple



$$d(2,4) = 2$$

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

$$\text{diam}(G) = d(2,7) = 4$$

Proposition

Soit G un GNO. Les assertions suivantes sont équivalentes.

1. G est connexe et sans cycle.
2. Pour tous sommets x et y , il existe une unique chaîne reliant x et y .
3. G est connexe, et ne l'est plus en retirant une arête.
4. G est sans cycle, et on crée un cycle en ajoutant une arête.
5. G est connexe et $|E| = |V| - 1$.
6. G est sans cycle et $|E| = |V| - 1$.

Proposition

Soit G un GNO. Les assertions suivantes sont équivalentes.

1. G est connexe et sans cycle.
2. Pour tous sommets x et y , il existe une unique chaîne reliant x et y .
3. G est connexe, et ne l'est plus en retirant une arête.
4. G est sans cycle, et on crée un cycle en ajoutant une arête.
5. G est connexe et $|E| = |V| - 1$.
6. G est sans cycle et $|E| = |V| - 1$.

Démonstration.

On montre $1 \Leftrightarrow 5 \Leftrightarrow 6$ par récurrence sur $n = |V|$.

Si $n = 1$, les trois assertions sont bien équivalentes.

Soit $n \geq 2$. Supposons les assertions équivalentes pour tous les graphes d'ordre strictement inférieur à n . □

Proposition

Soit G un GNO. Les assertions suivantes sont équivalentes.

1. G est connexe et sans cycle.
2. Pour tous sommets x et y , il existe une unique chaîne reliant x et y .
3. G est connexe, et ne l'est plus en retirant une arête.
4. G est sans cycle, et on crée un cycle en ajoutant une arête.
5. G est connexe et $|E| = |V| - 1$.
6. G est sans cycle et $|E| = |V| - 1$.

Démonstration.

$1 \Rightarrow 6$: supprimons une arête. Cela déconnecte G (sinon il y aurait un cycle). Les deux composantes obtenues sont connexes et sans cycle, d'ordre inférieur à n . Par HR :

$$|E| = |E_1| + |E_2| + 1 = |V_1| - 1 + |V_2| - 1 + 1 = |V| - 1. \quad \square$$

Proposition

Soit G un GNO. Les assertions suivantes sont équivalentes.

1. G est connexe et sans cycle.
2. Pour tous sommets x et y , il existe une unique chaîne reliant x et y .
3. G est connexe, et ne l'est plus en retirant une arête.
4. G est sans cycle, et on crée un cycle en ajoutant une arête.
5. G est connexe et $|E| = |V| - 1$.
6. G est sans cycle et $|E| = |V| - 1$.

Démonstration.

$6 \Rightarrow 5$: notons G_1, \dots, G_k ses composantes connexes. Elles sont connexes, sans cycle car G l'est, et d'ordres inférieurs à n . Par HR : $|E| = \sum |E_i| = \sum |V_i| - k = |V| - k$. Or $|E| = |V| - 1$. Donc $k = 1$ et G est connexe ! □

Proposition

Soit G un GNO. Les assertions suivantes sont équivalentes.

1. G est connexe et sans cycle.
2. Pour tous sommets x et y , il existe une unique chaîne reliant x et y .
3. G est connexe, et ne l'est plus en retirant une arête.
4. G est sans cycle, et on crée un cycle en ajoutant une arête.
5. G est connexe et $|E| = |V| - 1$.
6. G est sans cycle et $|E| = |V| - 1$.

Démonstration.

$5 \Rightarrow 1$: G a un sommet pendant, sinon

$\sum_{x \in V} d(x) = 2|E| = 2|V| - 2 \geq 2|V|$. Le sous-graphe induit G' en retirant

ce sommet vérifie HR. G' est donc sans cycle, tout comme G .



Proposition

Soit G un GNO. Les assertions suivantes sont équivalentes.

1. G est connexe et sans cycle.
2. Pour tous sommets x et y , il existe une unique chaîne reliant x et y .
3. G est connexe, et ne l'est plus en retirant une arête.
4. G est sans cycle, et on crée un cycle en ajoutant une arête.
5. G est connexe et $|E| = |V| - 1$.
6. G est sans cycle et $|E| = |V| - 1$.

Démonstration.

$1 \Rightarrow 3$: Supposons que G soit connexe en retirant un arête $e = xy$. Par connexité, il existe une chaîne reliant x et y . On aurait donc un cycle, ce qui est exclu.



Proposition

Soit G un GNO. Les assertions suivantes sont équivalentes.

1. G est connexe et sans cycle.
2. Pour tous sommets x et y , il existe une unique chaîne reliant x et y .
3. G est connexe, et ne l'est plus en retirant une arête.
4. G est sans cycle, et on crée un cycle en ajoutant une arête.
5. G est connexe et $|E| = |V| - 1$.
6. G est sans cycle et $|E| = |V| - 1$.

Démonstration.

$3 \Rightarrow 2$: Soit $\{x, y\} \subset V$. Il existe une chaîne de x à y car G est connexe. Supposons par l'absurde qu'il y en a une autre. Alors il y a un cycle. En retirant une arête de ce cycle, G reste connexe! \square

Proposition

Soit G un GNO. Les assertions suivantes sont équivalentes.

1. G est connexe et sans cycle.
2. Pour tous sommets x et y , il existe une unique chaîne reliant x et y .
3. G est connexe, et ne l'est plus en retirant une arête.
4. G est sans cycle, et on crée un cycle en ajoutant une arête.
5. G est connexe et $|E| = |V| - 1$.
6. G est sans cycle et $|E| = |V| - 1$.

Démonstration.

$2 \Rightarrow 4$: si G admettait un cycle, deux sommets de ce cycle seraient reliés par 2 chaînes ; donc G est sans cycle.

Si x et y ne sont pas adjacents ; il y a une unique chaîne de x vers y .
Mais en ajoutant l'arête xy on crée une deuxième chaîne. \square

Proposition

Soit G un GNO. Les assertions suivantes sont équivalentes.

1. G est connexe et sans cycle.
2. Pour tous sommets x et y , il existe une unique chaîne reliant x et y .
3. G est connexe, et ne l'est plus en retirant une arête.
4. G est sans cycle, et on crée un cycle en ajoutant une arête.
5. G est connexe et $|E| = |V| - 1$.
6. G est sans cycle et $|E| = |V| - 1$.

Définition

Un GNO qui vérifie l'une quelconque des assertions précédentes est appelé un *arbre*.

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

Peut-on trouver un cycle passant une fois et une seule par chaque arête ?
Un tel cycle est appelé *eulérien*.

Peut-on trouver un cycle passant une fois et une seule par chaque sommet ? Un tel cycle est appelé *hamiltonien*.

Remarque : le terme "cycle eulérien" ne correspond pas à notre définition du cycle élémentaire. Il faudrait parler de parcours, mais ce n'est pas grave.

Définitions

- Un graphe est *eulérien* (resp. *hamiltonien*) s'il admet un cycle eulérien (resp. hamiltonien).

Définitions

- Un graphe est *eulérien* (resp. *hamiltonien*) s'il admet un cycle eulérien (resp. hamiltonien).
- Un graphe est *semi-eulérien* (resp. *semi-hamiltonien*) s'il admet une chaîne eulérienne (resp. hamiltonienne).

Définitions

- Un graphe est *eulérien* (resp. *hamiltonien*) s'il admet un cycle eulérien (resp. hamiltonien).
- Un graphe est *semi-eulérien* (resp. *semi-hamiltonien*) s'il admet une chaîne eulérienne (resp. hamiltonienne).

Existe-t-il des conditions nécessaires et suffisantes « simples » pour justifier qu'un graphe est eulérien / hamiltonien ?

Théorème (Euler 1736)

- *Un graphe est eulérien SSI il est connexe et tous ses sommets sont de degré pair.*

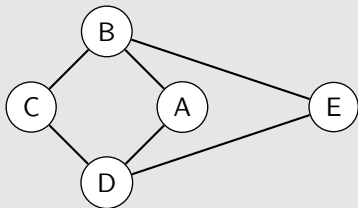
Théorème (Euler 1736)

- *Un graphe est eulérien SSI il est connexe et tous ses sommets sont de degré pair.*
- *Un graphe est semi-eulérien SSI il est connexe et tous ses sommets, sauf deux, sont de degré pair.*

Théorème (Euler 1736)

- *Un graphe est eulérien SSI il est connexe et tous ses sommets sont de degré pair.*
- *Un graphe est semi-eulérien SSI il est connexe et tous ses sommets, sauf deux, sont de degré pair.*

Exemple

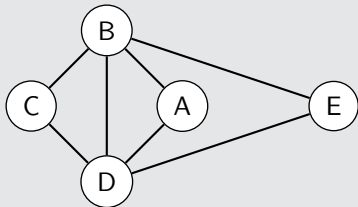


semi-eulérien

Théorème (Euler 1736)

- *Un graphe est eulérien SSI il est connexe et tous ses sommets sont de degré pair.*
- *Un graphe est semi-eulérien SSI il est connexe et tous ses sommets, sauf deux, sont de degré pair.*

Exemple



eulérien

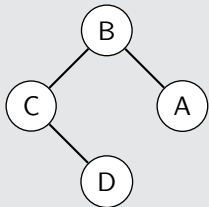
Hamiltonien, une tout autre affaire. . .

Problème NP-complet, on ne connaît pas de « bonne » condition (vérifiable en temps polynomial).

Hamiltonien, une tout autre affaire...

Problème NP-complet, on ne connaît pas de « bonne » condition (vérifiable en temps polynomial).

Exemple

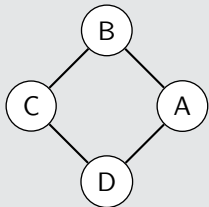


semi-hamiltonien

Hamiltonien, une tout autre affaire...

Problème NP-complet, on ne connaît pas de « bonne » condition (vérifiable en temps polynomial).

Exemple



hamiltonien

Quelques conditions suffisantes (mais non nécessaires) :

Quelques conditions suffisantes (mais non nécessaires) :

Théorème (Dirac 1952)

Si, dans un graphe d'ordre $n \geq 3$, tous les sommets sont de degré supérieur ou égal à $n/2$, alors le graphe est hamiltonien.

Quelques conditions suffisantes (mais non nécessaires) :

Théorème (Dirac 1952)

Si, dans un graphe d'ordre $n \geq 3$, tous les sommets sont de degré supérieur ou égal à $n/2$, alors le graphe est hamiltonien.

Théorème (Ore 1960)

Si, dans un graphe d'ordre $n \geq 3$, pour toute paire de sommets $\{x, y\}$ non adjacents, on a $d(x) + d(y) \geq n$, alors le graphe est hamiltonien.

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

On veut parcourir (ou déterminer) une composante connexe en numérotant les sommets. Deux parcours classiques :

- parcours en largeur (BFS : *breadth-first search*)
- parcours en profondeur (DFS : *depth-first search*)

Parcours en largeur

Données : un graphe $G = (V, E)$ et un sommet $x_0 \in V$

Résultat : une numérotation α en largeur de la composante connexe contenant x_0 , et les distances d depuis x_0

début

FILE $\leftarrow x_0$;

$i \leftarrow 1$;

$\alpha(x_0) \leftarrow 1$;

$dist(x_0) \leftarrow 0$;

tant que FILE $\neq \emptyset$ **faire**

$x \leftarrow$ DEFILER ;

pour chaque $y \in N(x)$ non numéroté **faire**

$i \leftarrow i + 1$;

$\alpha(y) \leftarrow i$;

$dist(y) \leftarrow dist(x) + 1$;

 ENFILER(y) ;

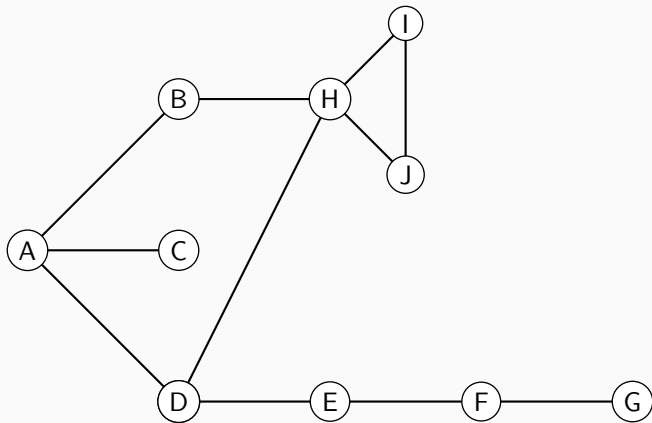
fin

fin

retourner $\alpha, dist$;

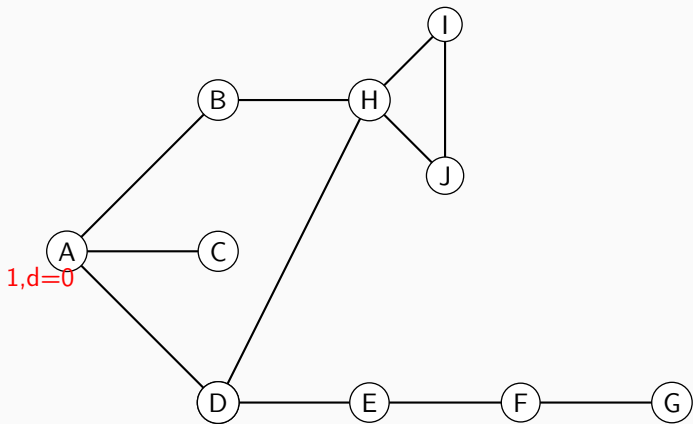
fin

Exemple de BFS



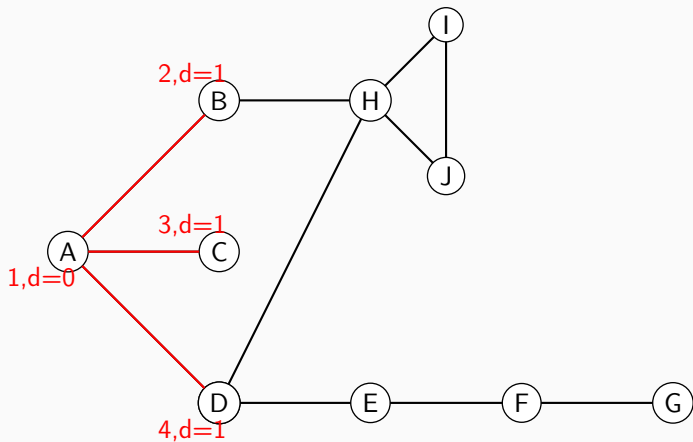
FILE : vide

Exemple de BFS



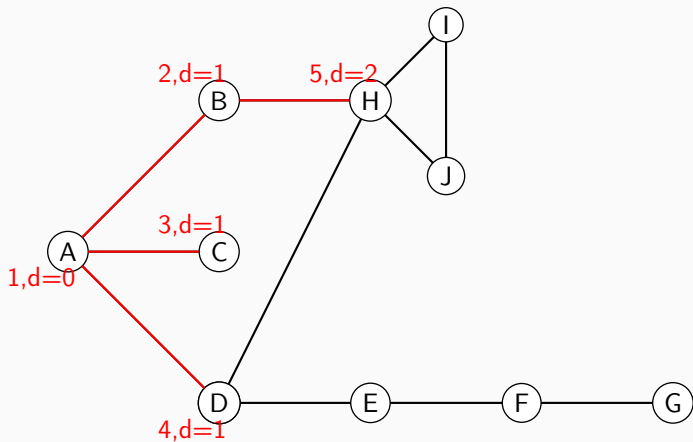
FILE : A

Exemple de BFS



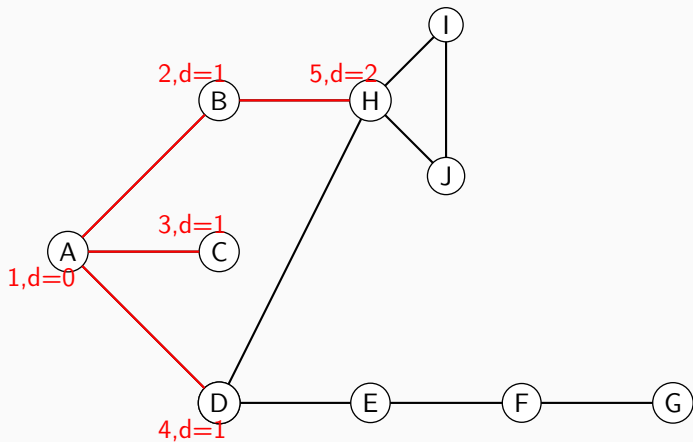
FILE : B C D

Exemple de BFS



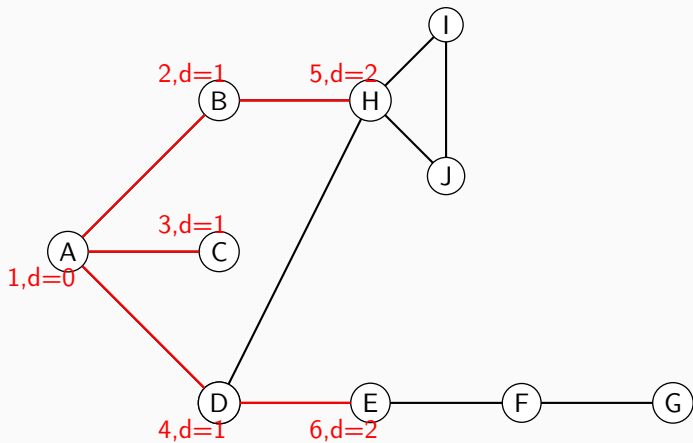
FILE : C D H

Exemple de BFS



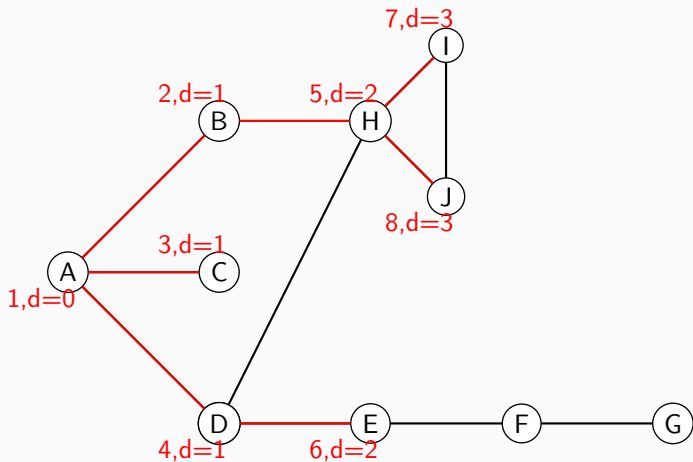
FILE : D H

Exemple de BFS



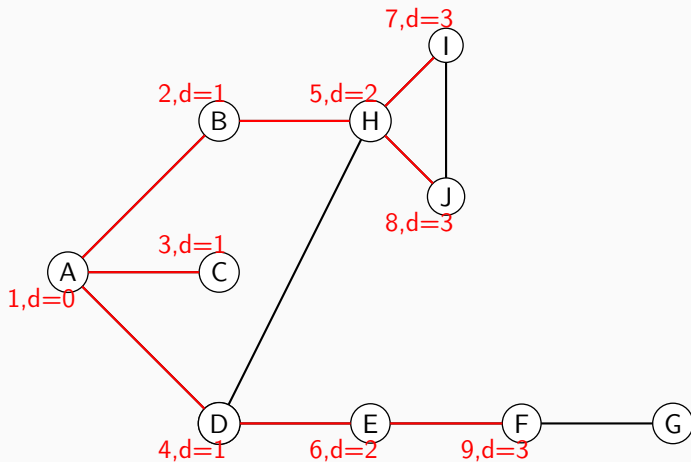
FILE : H E

Exemple de BFS



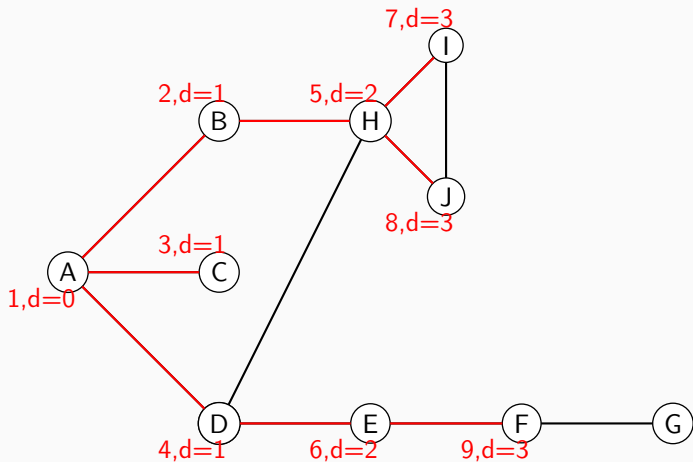
FILE : E I J

Exemple de BFS



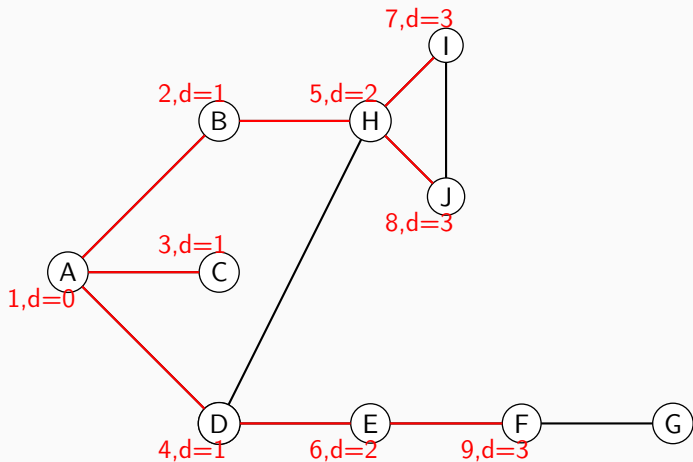
FILE : I J F

Exemple de BFS



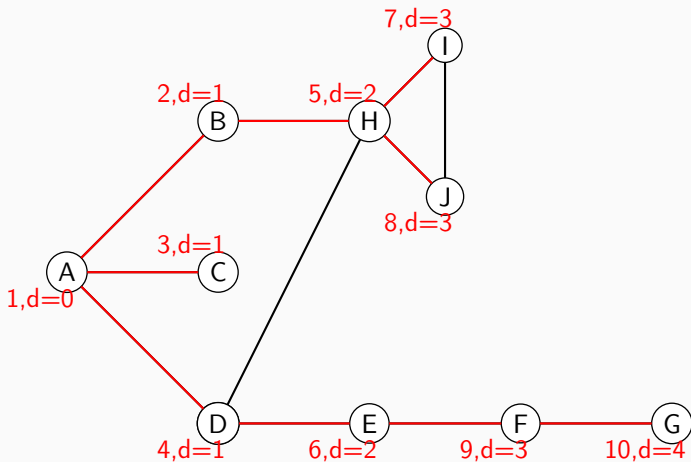
FILE : J F

Exemple de BFS



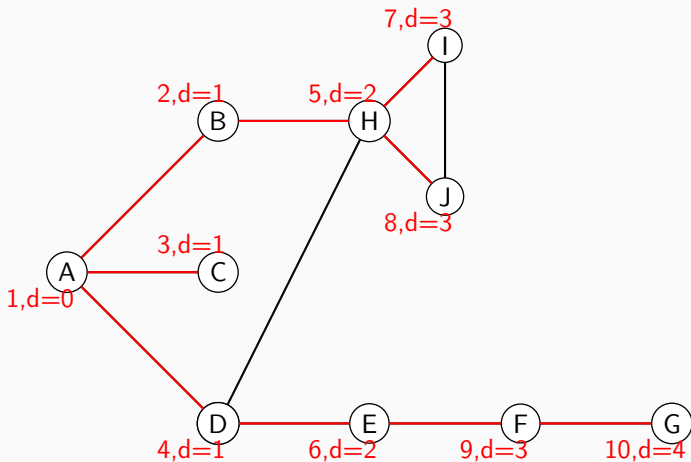
FILE : F

Exemple de BFS



FILE : G

Exemple de BFS



FILE : vide

L'algorithme fournit :

- une partition des sommets en couches (couche i = sommets à distance i du sommet initial)
- une chaîne de longueur minimal entre le sommet initial et les autres sommets.

Parcours en profondeur

Données : un graphe $G = (V, E)$ et un sommet $x_0 \in V$

Résultat : une numérotation α en profondeur de la composante connexe contenant x_0

début

PILE $\leftarrow x_0$;

$i \leftarrow 1$;

$\alpha(x_0) \leftarrow i$;

tant que PILE $\neq \emptyset$ **faire**

si le sommet de la pile a un voisin x pas encore visité **alors**

$i \leftarrow i + 1$;

$\alpha(x) \leftarrow i$;

 EMPILER(x) ;

sinon

$x \leftarrow$ DEPILER ;

fin

fin

retourner α ;

fin

Parcours en profondeur récursif

Données : un graphe $G = (V, E)$ et un sommet $x_0 \in V$

Résultat : une numérotation α en profondeur de la composante connexe contenant x_0

début

```
 $i \leftarrow 0 ;$ 
```

```
Fonction  $DFS\_rec(\text{graphe } G, \text{sommet } s)$ 
```

```
begin
```

```
   $i \leftarrow i + 1 ;$ 
```

```
   $\alpha(s) \leftarrow i ;$ 
```

```
  pour chaque  $y \in N(s)$  non numéroté faire
```

```
     $DFS\_rec(G, y) ;$ 
```

```
  fin
```

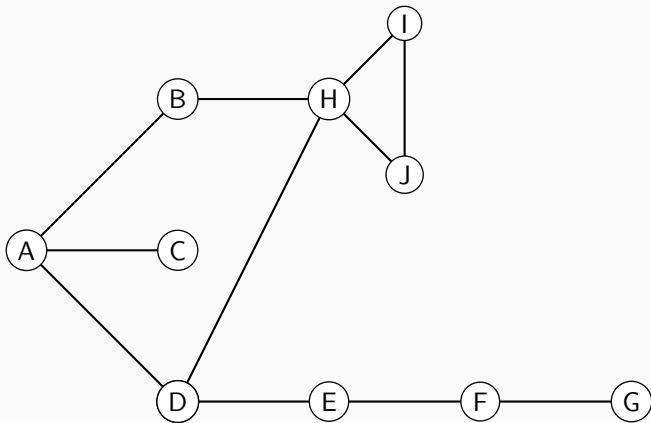
```
end
```

```
 $DFS\_rec(G, x_0) ;$ 
```

```
retourner  $\alpha ;$ 
```

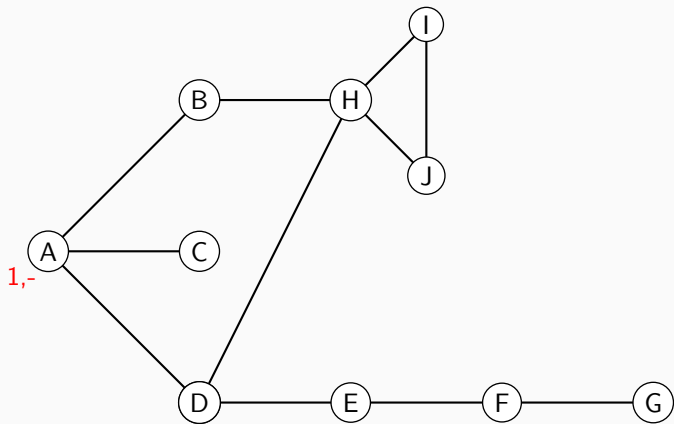
fin

Exemple de DFS



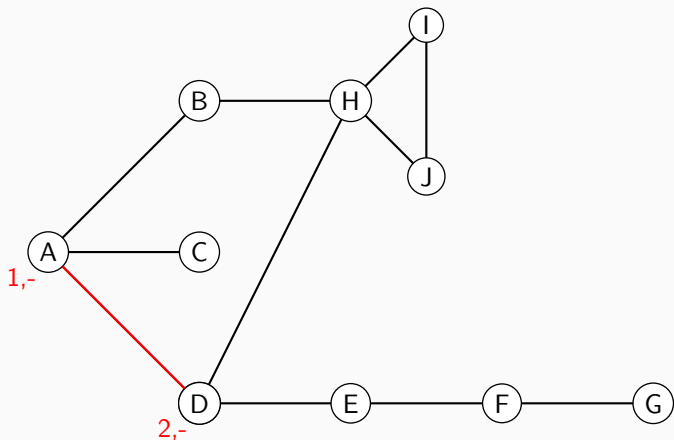
PILE : vide

Exemple de DFS



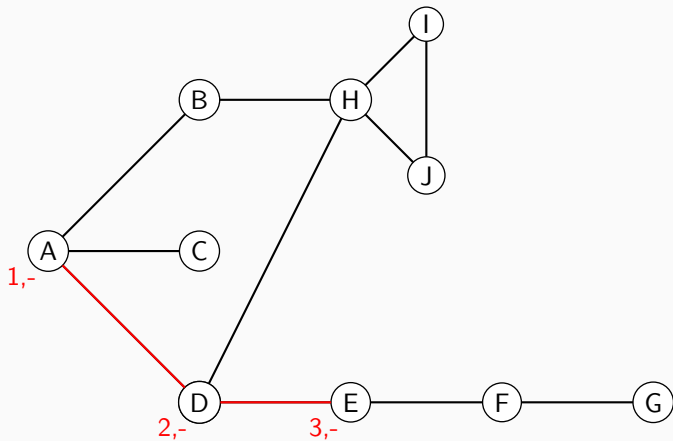
PILE : A

Exemple de DFS



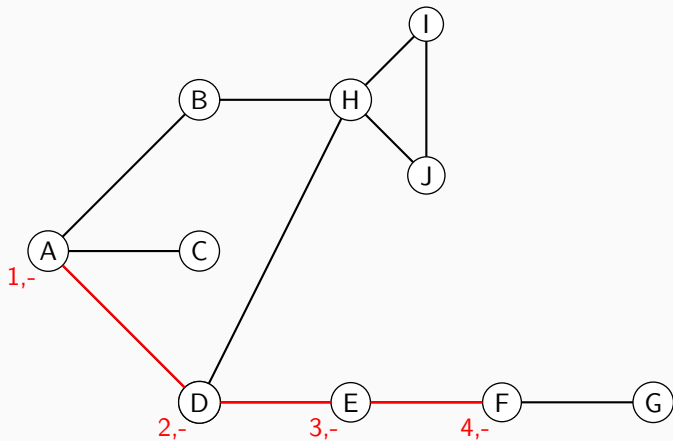
PILE : A D

Exemple de DFS



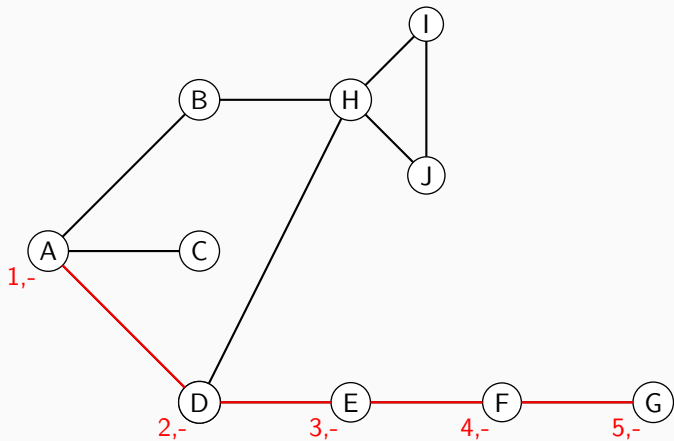
PILE : A D E

Exemple de DFS



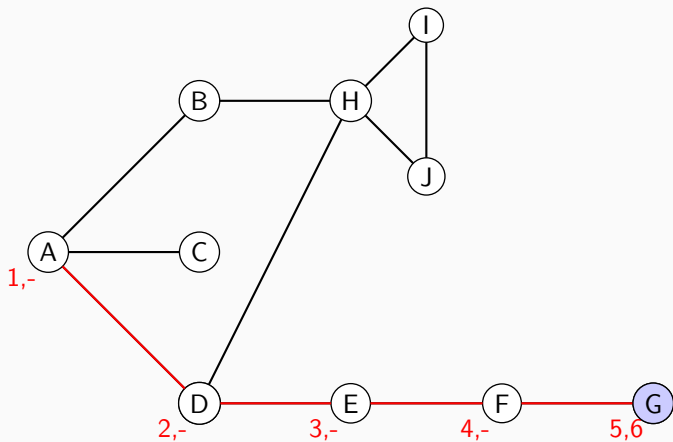
PILE : A D E F

Exemple de DFS



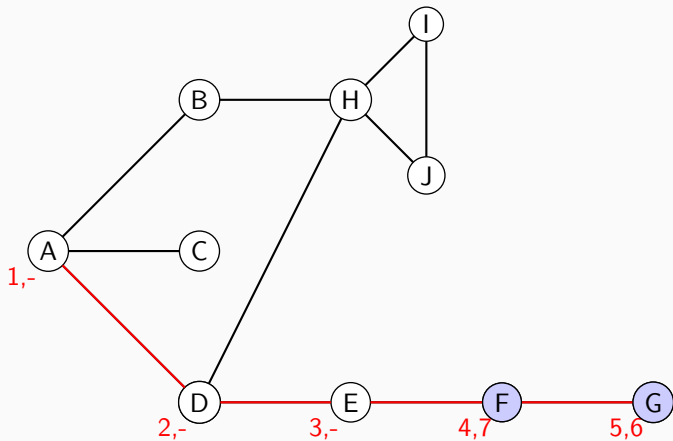
PILE : A D E F G

Exemple de DFS



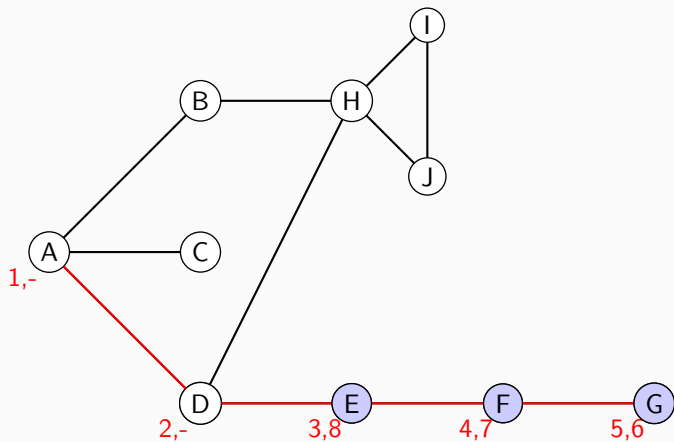
PILE : A D E F

Exemple de DFS



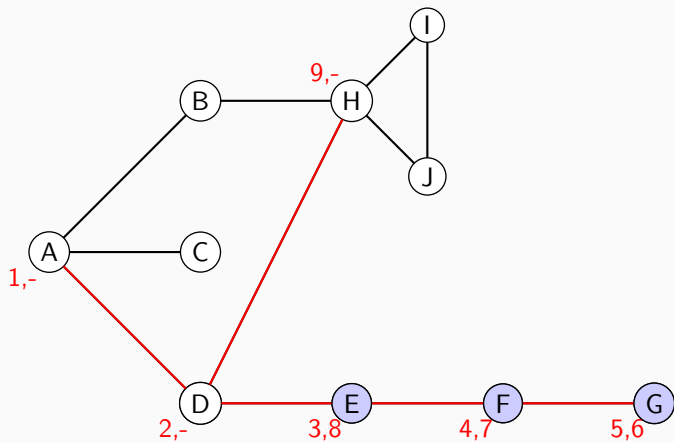
PILE : A D E

Exemple de DFS



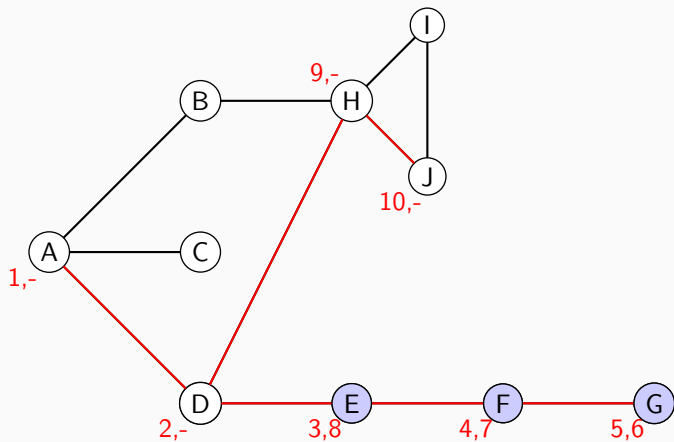
PILE : A D

Exemple de DFS



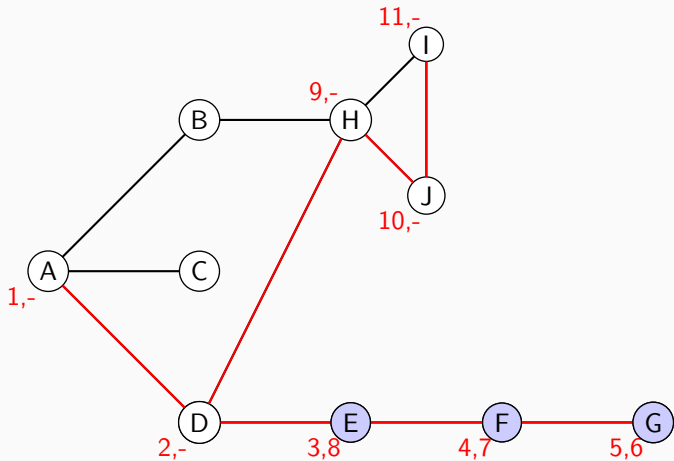
PILE : A D H

Exemple de DFS



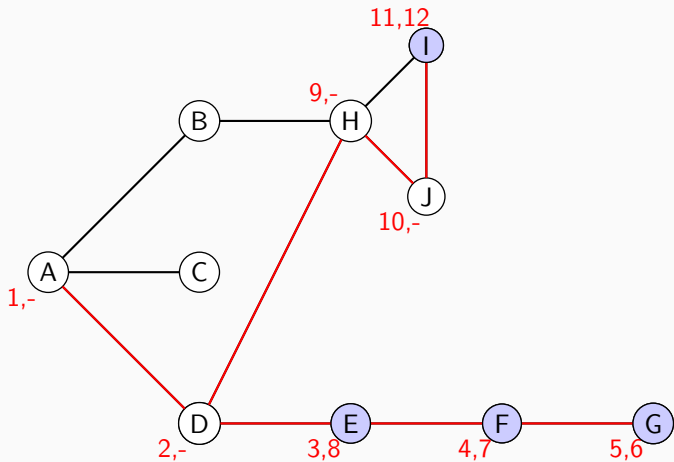
PILE : A D H J

Exemple de DFS



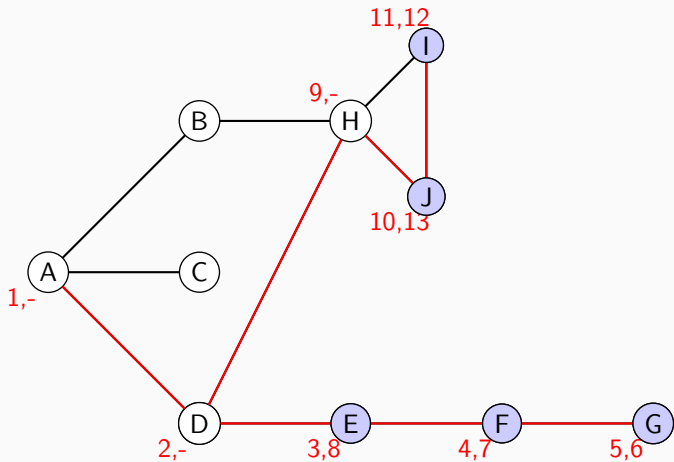
PILE : A D H J I

Exemple de DFS



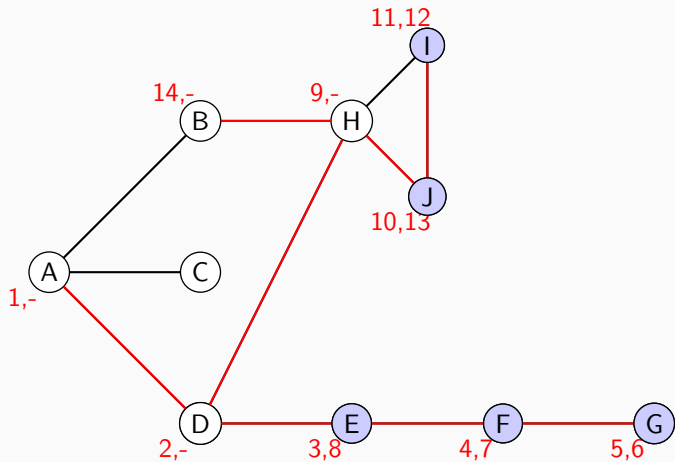
PILE : A D H J

Exemple de DFS



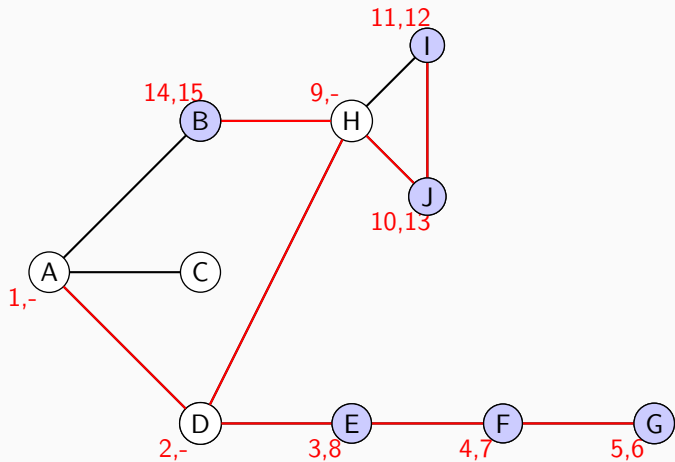
PILE : A D H

Exemple de DFS



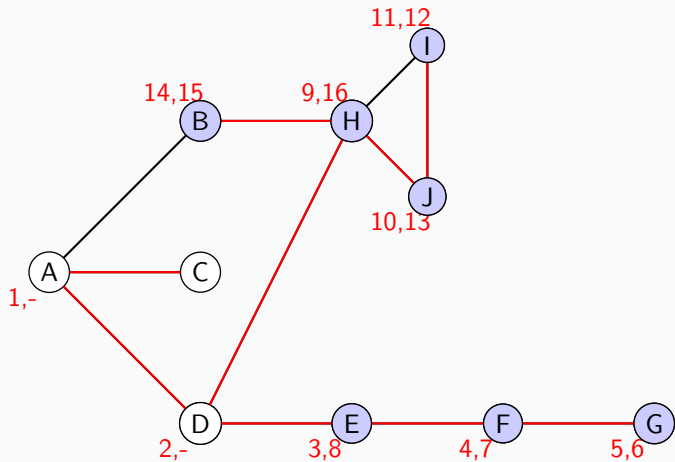
PILE : A D H B

Exemple de DFS



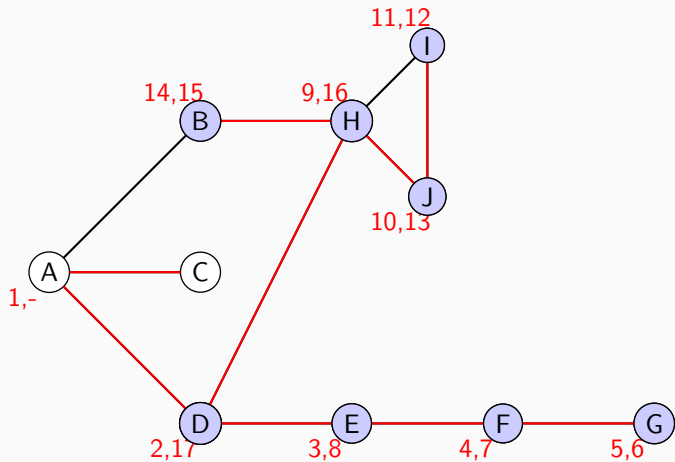
PILE : A D H

Exemple de DFS



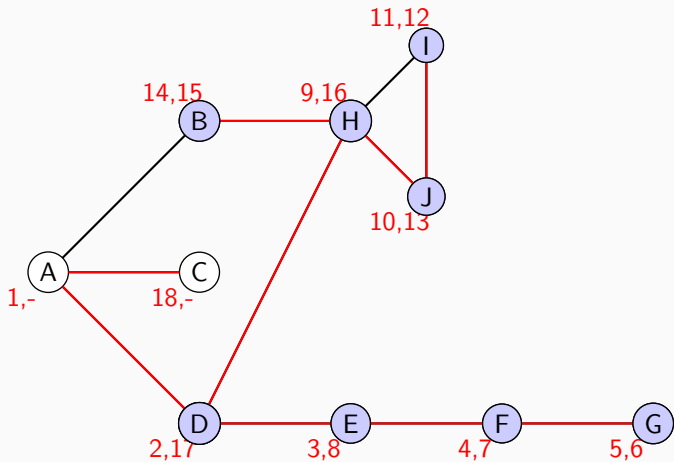
PILE : A D

Exemple de DFS



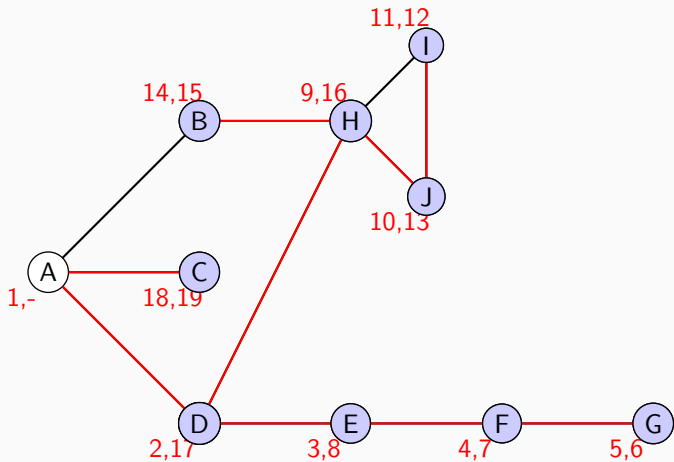
PILE : A

Exemple de DFS



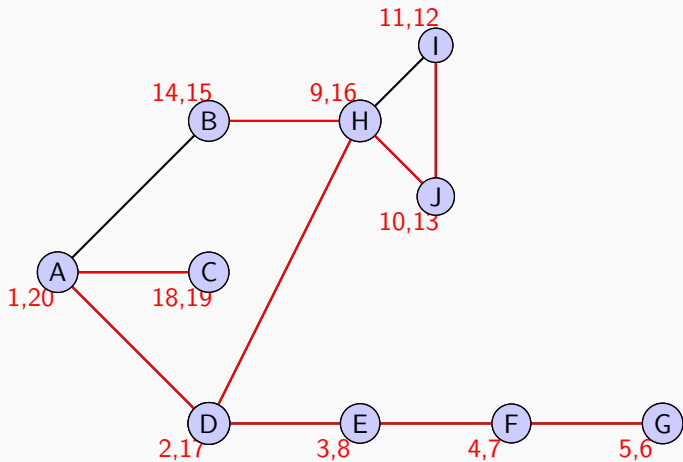
PILE : A C

Exemple de DFS



PILE : A

Exemple de DFS



PILE : vide

Soit $G = (V, E)$ un *GO*

Un tri topologique de G est la donnée d'une injection $f : E \rightarrow \mathbb{N}$ telle que

$$\forall (x, y) \in V, xy \in E \Rightarrow f(x) < f(y)$$

La numérotation des sommets est compatible avec les arcs de G .

Proposition

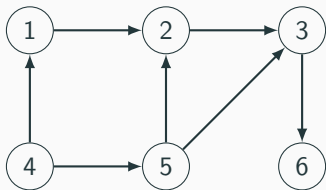
Un graphe admet un tri topologique ssi il est sans circuit.

Proposition

Les dates de fin de visites d'un parcours en profondeur d'un *GO acyclique* fournissent un tri topologique.

Exemple

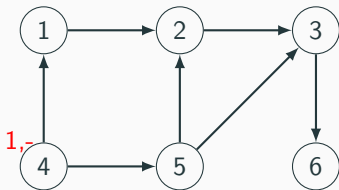
DFS en partant du sommet 4



Tri topologique :

Exemple

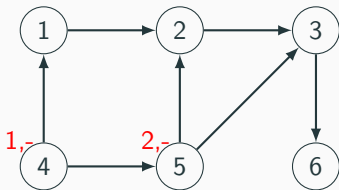
DFS en partant du sommet 4



Tri topologique :

Exemple

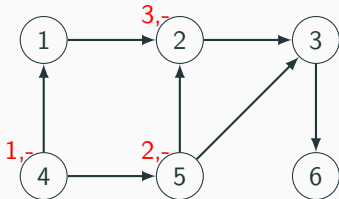
DFS en partant du sommet 4



Tri topologique :

Exemple

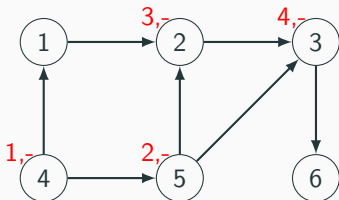
DFS en partant du sommet 4



Tri topologique :

Exemple

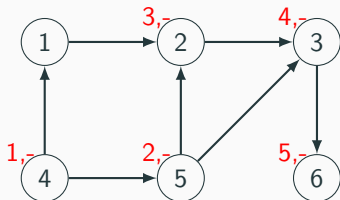
DFS en partant du sommet 4



Tri topologique :

Exemple

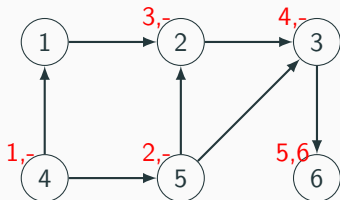
DFS en partant du sommet 4



Tri topologique :

Exemple

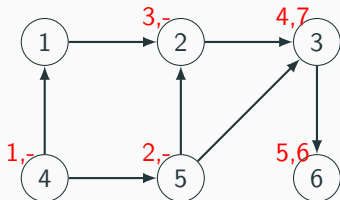
DFS en partant du sommet 4



Tri topologique : 6

Exemple

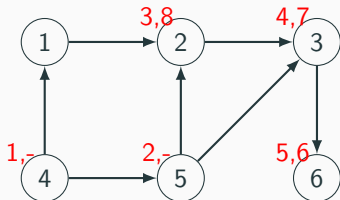
DFS en partant du sommet 4



Tri topologique : 3 6

Exemple

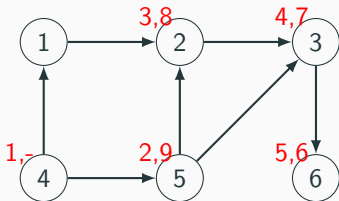
DFS en partant du sommet 4



Tri topologique : 2 3 6

Exemple

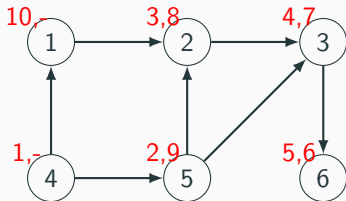
DFS en partant du sommet 4



Tri topologique : 5 2 3 6

Exemple

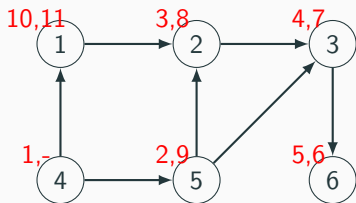
DFS en partant du sommet 4



Tri topologique : 5 2 3 6

Exemple

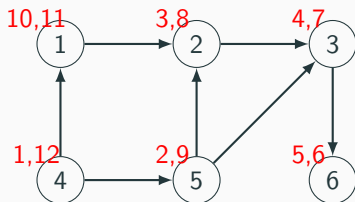
DFS en partant du sommet 4



Tri topologique : 1 5 2 3 6

Exemple

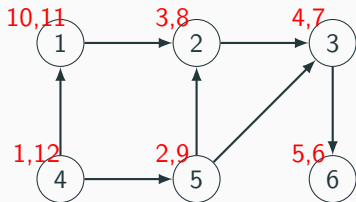
DFS en partant du sommet 4



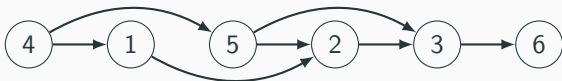
Tri topologique : 4 1 5 2 3 6

Exemple

DFS en partant du sommet 4

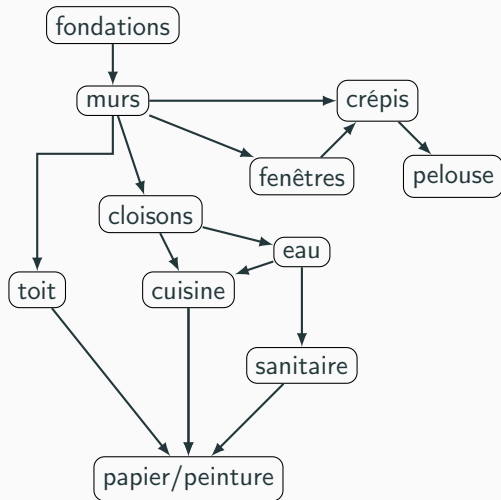


Tri topologique : 4 1 5 2 3 6



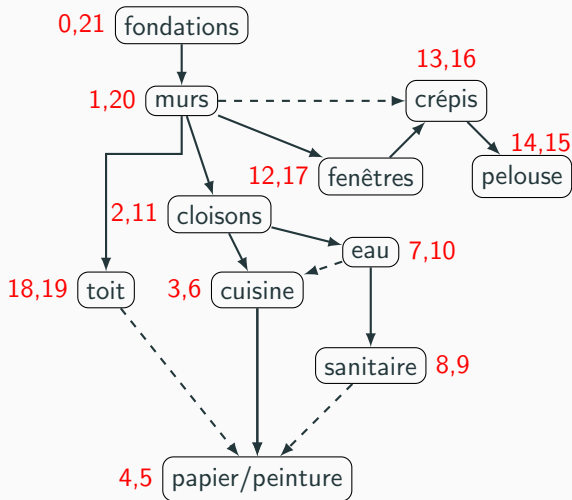
Application à l'ordonnancement de tâches

Construire une maison. Arc $x \rightarrow y$: "commencer y que si x est terminé"



Application à l'ordonancement de tâches

Construire une maison. Arc $x \rightarrow y$: "commencer y que si x est terminé"



Ordre possible :

fondations	21
murs	20
toit	19
fenêtres	17
crépis	16
pelouse	15
cloisons	11
eau	10
sanitaires	9
cuisine	6
papiers/peinture	5

Graphes et couleurs

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

Colorier un graphe, c'est attribuer une couleur à chaque sommet, de sorte que deux sommets adjacents aient une couleur différente.

But : utiliser le moins de couleurs

Colorier un graphe, c'est attribuer une couleur à chaque sommet, de sorte que deux sommets adjacents aient une couleur différente.

But : utiliser le moins de couleurs

Définition

On appelle *nombre chromatique* d'un graphe G , noté $\chi(G)$, le plus petit nombre de couleurs nécessaires à la coloration de G .

Colorier un graphe, c'est attribuer une couleur à chaque sommet, de sorte que deux sommets adjacents aient une couleur différente.

But : utiliser le moins de couleurs

Définition

On appelle *nombre chromatique* d'un graphe G , noté $\chi(G)$, le plus petit nombre de couleurs nécessaires à la coloration de G .

Exemples

$$\chi(K_n) = n$$

$$\chi(P_n) = 2$$

$$\chi(C_{2n}) = 2$$

$$\chi(C_{2n+1}) = 3$$

Colorier un graphe, c'est attribuer une couleur à chaque sommet, de sorte que deux sommets adjacents aient une couleur différente.

But : utiliser le moins de couleurs

Définition

On appelle *nombre chromatique* d'un graphe G , noté $\chi(G)$, le plus petit nombre de couleurs nécessaires à la coloration de G .

Exemples

$$\chi(K_n) = n$$

$$\chi(P_n) = 2$$

$$\chi(C_{2n}) = 2$$

$$\chi(C_{2n+1}) = 3$$

Pas facile en pratique sur un graphe quelconque. . .

On note $\omega(G)$ la taille d'une clique maximale et $\Delta(G)$ le degré maximum d'un de ses sommets.

On note $\omega(G)$ la taille d'une clique maximale et $\Delta(G)$ le degré maximum d'un de ses sommets.

Proposition

$$\omega(G) \leq \chi(G) \leq \Delta(G) + 1 \leq |V|$$

Encadrement de $\chi(G)$

On note $\omega(G)$ la taille d'une clique maximale et $\Delta(G)$ le degré maximum d'un de ses sommets.

Proposition

$$\omega(G) \leq \chi(G) \leq \Delta(G) + 1 \leq |V|$$

- Résultat général optimal car les quatre membres sont égaux si $G = K_n$;
- mais décevant car l'écart entre $\chi(G)$ et $\Delta(G)$ peut tendre vers l'infini ! (considérer S_n) ;
- $\omega(G)$ est généralement difficile à déterminer.

Algorithme de Welsh-Powell

algorithme heuristique (solution approchée) et glouton (donc efficace)

Données : un graphe $G = (V, E)$ non orienté

Résultat : une coloration α des sommets

début

$L \leftarrow$ liste des sommets par ordre décroissant des degrés ;

$couleur \leftarrow 1$;

tant que *des sommets ne sont pas coloriés* **faire**

$s \leftarrow$ premier sommet de L non colorié ;

$\alpha(s) \leftarrow couleur$;

pour $x \in L$ *non adjacent à s et non adjacent à un sommet de*
 couleur faire

$\alpha(x) \leftarrow couleur$;

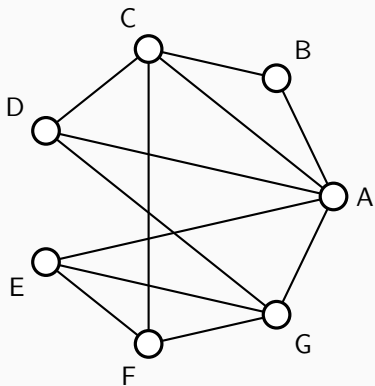
fin

$couleur \leftarrow couleur + 1$;

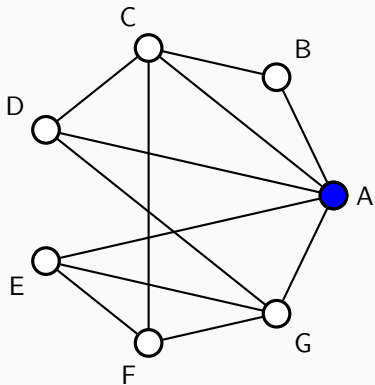
fin

retourner α ;

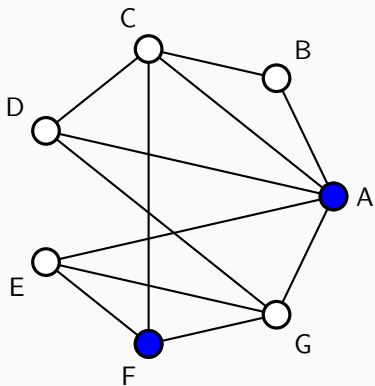
fin



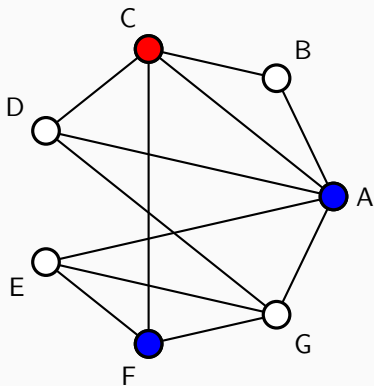
Sommet	Degré	Couleur
A	5	
C	4	
G	4	
D	3	
E	3	
F	3	
B	2	



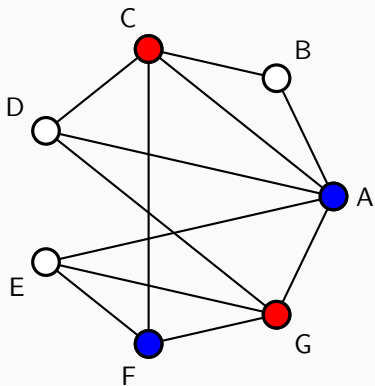
Sommet	Degré	Couleur
A	5	bleu
C	4	
G	4	
D	3	
E	3	
F	3	
B	2	



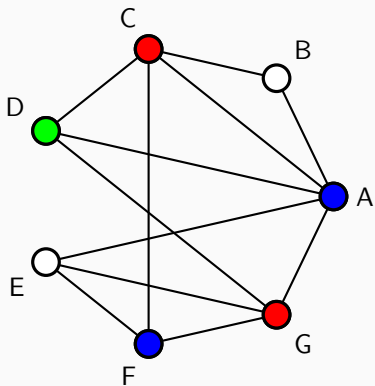
Sommet	Degré	Couleur
A	5	bleu
C	4	
G	4	
D	3	
E	3	
F	3	bleu
B	2	



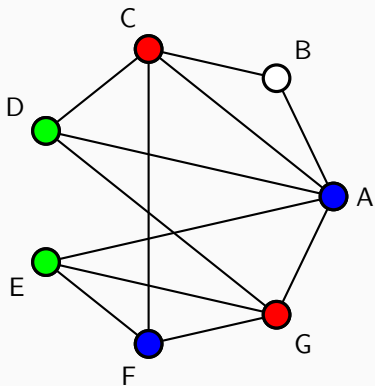
Sommet	Degré	Couleur
A	5	bleu
C	4	rouge
G	4	
D	3	
E	3	
F	3	bleu
B	2	



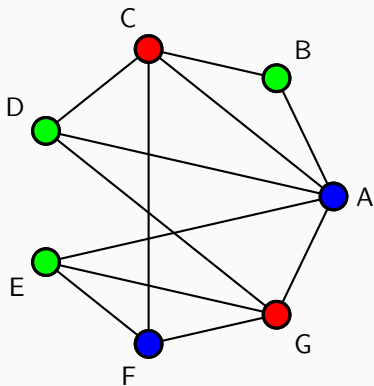
Sommet	Degré	Couleur
A	5	bleu
C	4	rouge
G	4	rouge
D	3	
E	3	
F	3	bleu
B	2	



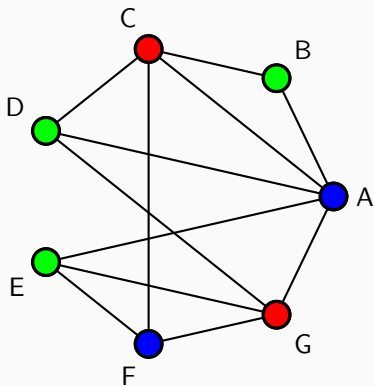
Sommet	Degré	Couleur
A	5	bleu
C	4	rouge
G	4	rouge
D	3	vert
E	3	
F	3	bleu
B	2	



Sommet	Degré	Couleur
A	5	bleu
C	4	rouge
G	4	rouge
D	3	vert
E	3	vert
F	3	bleu
B	2	

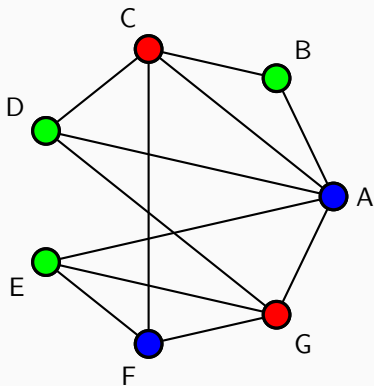


Sommet	Degré	Couleur
A	5	bleu
C	4	rouge
G	4	rouge
D	3	vert
E	3	vert
F	3	bleu
B	2	vert



Sommet	Degré	Couleur
A	5	bleu
C	4	rouge
G	4	rouge
D	3	vert
E	3	vert
F	3	bleu
B	2	vert

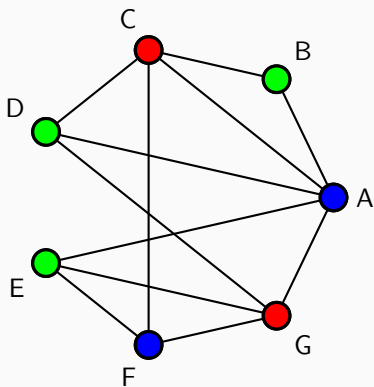
Ainsi $\chi(G) \leq 3$.



Sommet	Degré	Couleur
A	5	bleu
C	4	rouge
G	4	rouge
D	3	vert
E	3	vert
F	3	bleu
B	2	vert

Ainsi $\chi(G) \leq 3$.

Or $\omega(G) \geq 3$ car G contient des triangles.



Sommet	Degré	Couleur
A	5	bleu
C	4	rouge
G	4	rouge
D	3	vert
E	3	vert
F	3	bleu
B	2	vert

Ainsi $\chi(G) \leq 3$.

Or $\omega(G) \geq 3$ car G contient des triangles.

Donc $\chi(G) = 3$.

La coloration de graphe permet de résoudre des problèmes divers :

- tâches à effectuer mais certaines ne pouvant se faire simultanément (gestion d'emplois du temps...);
- créer des groupes en respectant des incompatibilités;
- théorème de la galerie d'art (voir TD).

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

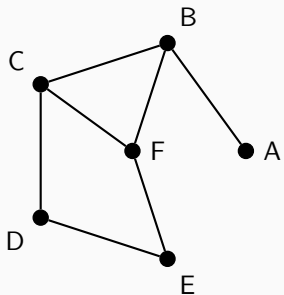
Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

Formule d'Euler

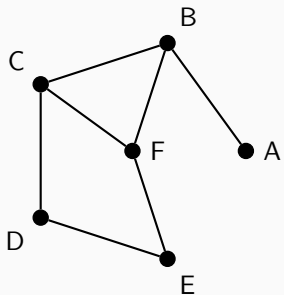


Nombre de sommets : $n = 6$

Nombre d'arêtes : $m = 7$

Nombre de faces : $f = 3$

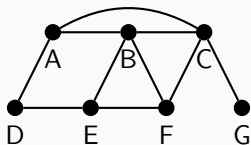
Formule d'Euler



Nombre de sommets : $n = 6$

Nombre d'arêtes : $m = 7$

Nombre de faces : $f = 3$

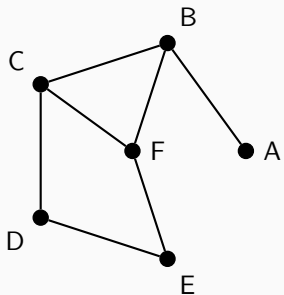


Nombre de sommets : $n = 7$

Nombre d'arêtes : $m = 10$

Nombre de faces : $f = 5$

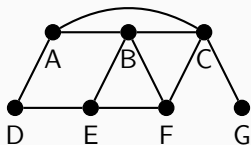
Formule d'Euler



Nombre de sommets : $n = 6$

Nombre d'arêtes : $m = 7$

Nombre de faces : $f = 3$



Nombre de sommets : $n = 7$

Nombre d'arêtes : $m = 10$

Nombre de faces : $f = 5$

On constate que $n + f - m$ semble valoir toujours 2.

Théorème

Tout représentation plane d'un graphe G planaire connexe vérifie la formule d'Euler : $n + f - m = 2$.

Démonstration.



Théorème

Tout représentation plane d'un graphe G planaire connexe vérifie la formule d'Euler : $n + f - m = 2$.

Démonstration.

Par récurrence sur m .

Si $m = 1$, alors par connexité $G = K_2$, donc $n = 2$ et $f = 1$. La formule d'Euler est bien vérifiée.



Théorème

Tout représentation plane d'un graphe G planaire connexe vérifie la formule d'Euler : $n + f - m = 2$.

Démonstration.

Supposons le théorème vrai pour les graphes ayant moins de $m \in \mathbb{N}^*$ arêtes. Soit G un graphe planaire connexe avec m arêtes (n sommets et f faces) :



Théorème

Tout représentation plane d'un graphe G planaire connexe vérifie la formule d'Euler : $n + f - m = 2$.

Démonstration.

Supposons le théorème vrai pour les graphes ayant moins de $m \in \mathbb{N}^*$ arêtes. Soit G un graphe planaire connexe avec m arêtes (n sommets et f faces) :

– soit G a un cycle ; en retirant un arête de ce cycle, le graphe reste planaire et connexe, avec n sommets, $m - 1$ arêtes et $f - 1$ faces. Par HR, on a $n + (f - 1) - (m - 1) = 2$. En développant on trouve $n + f - m = 2$.

Théorème

Tout représentation plane d'un graphe G planaire connexe vérifie la formule d'Euler : $n + f - m = 2$.

Démonstration.

Supposons le théorème vrai pour les graphes ayant moins de $m \in \mathbb{N}^*$ arêtes. Soit G un graphe planaire connexe avec m arêtes (n sommets et f faces) :

– soit G a un cycle ; en retirant un arête de ce cycle, le graphe reste planaire et connexe, avec n sommets, $m - 1$ arêtes et $f - 1$ faces. Par HR, on a $n + (f - 1) - (m - 1) = 2$. En développant on trouve $n + f - m = 2$.

– soit G n'a pas de cycle : c'est un arbre ; soit x un sommet pendant (il y en a au moins deux !). En retirant x et la seule arête issue de x , on obtient un graphe planaire connexe à $n - 1$ sommets, $m - 1$ arêtes et f faces. Par HR : $(n - 1) + f - (m - 1) = 2$. En développant on trouve $n + f - m = 2$. □

Théorème

1. *Dans un graphe planaire connexe avec $n > 2$, on a toujours $m \leq 3n - 6$.*
2. *Tout graphe planaire connexe admet au moins un sommet de degré au plus égal à 5.*

Théorème

1. Dans un graphe planaire connexe avec $n > 2$, on a toujours $m \leq 3n - 6$.
2. Tout graphe planaire connexe admet au moins un sommet de degré au plus égal à 5.

Démonstration.

1. Toute face est bordée par au moins 3 arêtes, et une arête appartient à au plus 2 faces, donc $3f \leq 2m$ soit $f \leq \frac{2}{3}m$. D'après la formule d'Euler, on a

$$m = n + f - 2 \leq n + \frac{2}{3}m - 2$$

d'où $m \leq 3n - 6$.

2. Par l'absurde, si tous les sommets étaient de degré au moins 6, la somme des degrés vaudrait au moins $6n$. Or cette somme vaut $2m$. Donc on aurait $m \geq 3n$, ce qui contredit le point précédent.



Théorème

K_5 n'est pas planaire.

Théorème

K_5 n'est pas planaire.

Démonstration.

Dans K_5 , on a $n = 5$, donc $3n - 6 = 9$. Mais $m = 10 > 3n - 6$. Donc K_5 a trop d'arêtes pour être planaire. \square

Théorème

$K_{3,3}$ n'est pas planaire.

Théorème

$K_{3,3}$ n'est pas planaire.

Démonstration.

Le même argument que pour K_5 ne fonctionne pas car $3n - 6 = 12$ et $m = 9$, donc $K_{3,3}$ n'a pas trop d'arête.



Théorème

$K_{3,3}$ n'est pas planaire.

Démonstration.

Le même argument que pour K_5 ne fonctionne pas car $3n - 6 = 12$ et $m = 9$, donc $K_{3,3}$ n'a pas trop d'arête.

Par l'absurde, supposons qu'il le soit. Il admet alors une représentation plane qui vérifie la formule d'Euler, d'où un nombre de faces

$$f = 2 - n + m = 2 - 6 + 9 = 5.$$

Cela fait en moyenne $\frac{2m}{f} = \frac{18}{5}$ arêtes par face. Ce nombre est inférieur à 4, alors qu'une face d'un graphe biparti est bordée par au moins 4 arêtes (pas de cycle de longueur impair). □

Théorème (Kuratowski 1930)

Un graphe est planaire SSI aucun de ses sous-graphes n'est une subdivision de K_5 ou $K_{3,3}$.

Théorème

Le nombre chromatique d'un graphe planaire est au plus 6.

Théorème

Le nombre chromatique d'un graphe planaire est au plus 6.

Démonstration.

Par récurrence sur n en utilisant le théorème précédent (sommet de degré au plus 5). □

Théorème

Le nombre chromatique d'un graphe planaire est au plus 6.

Démonstration.

Par récurrence sur n en utilisant le théorème précédent (sommet de degré au plus 5). □

Théorème (Heawood 1890)

Le nombre chromatique d'un graphe planaire est au plus 5.

Théorème

Le nombre chromatique d'un graphe planaire est au plus 6.

Démonstration.

Par récurrence sur n en utilisant le théorème précédent (sommet de degré au plus 5). □

Théorème (Heawood 1890)

Le nombre chromatique d'un graphe planaire est au plus 5.

Démonstration.

Par récurrence sur n en utilisant le théorème précédent (sommet de degré au plus 5) et une idée géniale. □

Théorème

Le nombre chromatique d'un graphe planaire est au plus 6.

Démonstration.

Par récurrence sur n en utilisant le théorème précédent (sommet de degré au plus 5). □

Théorème (Heawood 1890)

Le nombre chromatique d'un graphe planaire est au plus 5.

Démonstration.

Par récurrence sur n en utilisant le théorème précédent (sommet de degré au plus 5) et une idée géniale. □

Théorème (Appel & Haken 1976)

Le nombre chromatique d'un graphe planaire est au plus 4.

Graphes valués

Un graphe *valué* est un triplet $G = (V, E, f)$ où f est une fonction de E dans \mathbb{R} . Autrement dit chaque arête est munie d'une valeur.

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

Étant donné un graphe valué et un sommet initial $s \in V$, on cherche le plus court (de poids minimal) chemin de s aux autres sommets du graphe.

Deux algorithmes :

Étant donné un graphe valué et un sommet initial $s \in V$, on cherche le plus court (de poids minimal) chemin de s aux autres sommets du graphe.

Deux algorithmes :

- Dijkstra : uniquement avec des valuations positives
- Bellman-Ford : tous les graphes valués, mais sans cycle de poids total négatif

Algorithme de Dijkstra 1959

Données : un graphe $G = (V, E, p)$ pondéré par une fonction p , un sommet de départ $s \in V$

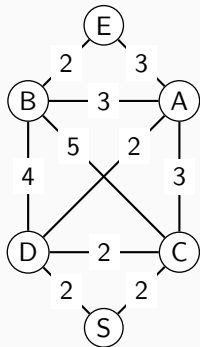
Résultat : une valuation d qui est la distance d'un PCC à partir de s ; une fonction $pere$ sur V donnant une arborescence représentant les PCC trouvés

début

```
TRAITE  $\leftarrow \emptyset$  ;  
pour  $x \in V$  faire  
|  $d(x) \leftarrow +\infty$  ;  
fin  
 $d(s) \leftarrow 0$  ;  
tant que TRAITE  $\neq V$  faire  
| choisir  $x \notin$  TRAITE tel que  $d(x)$  soit minimal ;  
| TRAITE  $\leftarrow x$  ;  
| pour  $y \in N(x) \setminus$  TRAITE faire  
| | si  $d(x) + p(xy) < d(y)$  alors  
| | |  $d(y) \leftarrow d(x) + p(xy)$  ;  
| | |  $pere(y) \leftarrow x$  ;  
| | fin  
| fin  
fin  
retourner  $d$  et  $pere$ 
```

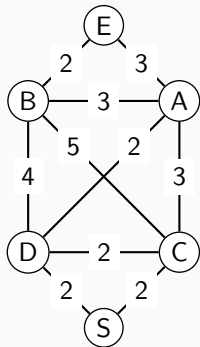
fin

Exemple Dijkstra depuis E



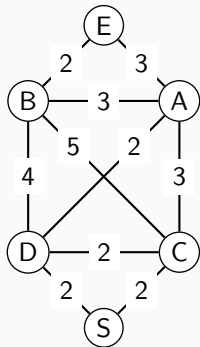
E	A	B	C	D	S	Traité
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	E

Exemple Dijkstra depuis E



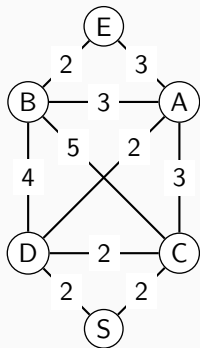
E	A	B	C	D	S	Traité
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	E
	3(E)	2(E)	$+\infty$	$+\infty$	$+\infty$	B

Exemple Dijkstra depuis E



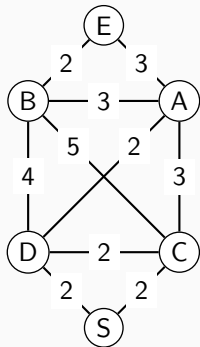
E	A	B	C	D	S	Traité
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	E
	3(E)	2(E)	$+\infty$	$+\infty$	$+\infty$	B
	3(E)		7(B)	6(B)	$+\infty$	A

Exemple Dijkstra depuis E



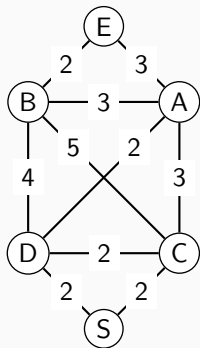
E	A	B	C	D	S	Traité
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	E
	3(E)	2(E)	$+\infty$	$+\infty$	$+\infty$	B
	3(E)		7(B)	6(B)	$+\infty$	A
			6(A)	5(A)	$+\infty$	D

Exemple Dijkstra depuis E



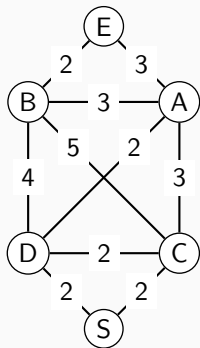
E	A	B	C	D	S	Traité
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	E
	3(E)	2(E)	$+\infty$	$+\infty$	$+\infty$	B
	3(E)		7(B)	6(B)	$+\infty$	A
			6(A)	5(A)	$+\infty$	D
			6(A)		7(D)	C

Exemple Dijkstra depuis E



E	A	B	C	D	S	Traité
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	E
	3(E)	2(E)	$+\infty$	$+\infty$	$+\infty$	B
	3(E)		7(B)	6(B)	$+\infty$	A
			6(A)	5(A)	$+\infty$	D
			6(A)		7(D)	C
					7(D)	S

Exemple Dijkstra depuis E



E	A	B	C	D	S	Traité
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	E
	3(E)	2(E)	$+\infty$	$+\infty$	$+\infty$	B
	3(E)		7(B)	6(B)	$+\infty$	A
			6(A)	5(A)	$+\infty$	D
			6(A)		7(D)	C
					7(D)	S

La dernière case de chaque colonne donne la distance minimale depuis E ainsi que le sommet d'où l'on vient, ce qui permet de reconstituer le trajet.

Par exemple, le plus court chemin de E vers S est de poids 7 : $E - A - D - S$.

Algorithme de Bellman-Ford 1956

Données : un graphe $G = (V, E, p)$ pondéré par une fonction p ; un sommet départ $s \in V$

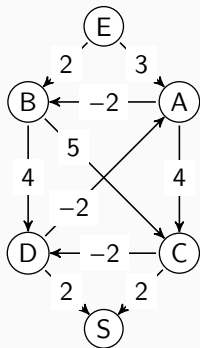
Résultat : une valuation d qui est la distance d'un PCC à partir de s ; une fonction $pere$ sur V donnant une arborescence représentant les PCC trouvés

début

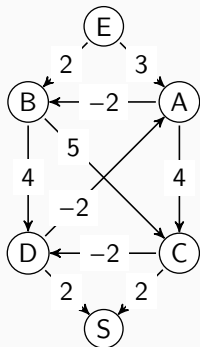
```
pour x ∈ V faire
    d(x,0) ← +∞ ;
    pere(x) ← NULL ;
fin
d(s,0) ← 0 ;
pour k de 1 à |V| - 1 faire
    pour xy ∈ E faire
        si d(x, k - 1) + p(xy) < d(y, k - 1) alors
            d(y, k) ← d(x, k - 1) + p(xy) ;
            pere(y) ← x ;
        fin
    fin
fin
// cycle de poids négatifs ?
pour xy ∈ E faire
    si d(x) + p(xy) < d(y) alors
        retourner ("Erreur : G contient un cycle de poids total
                    négatif")
    fin
fin
retourner d et pere
```

fin

Contre-exemple Bellman-Ford

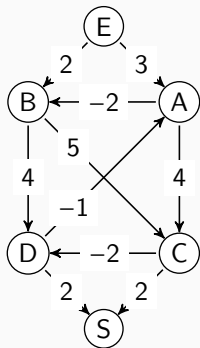


Contre-exemple Bellman-Ford



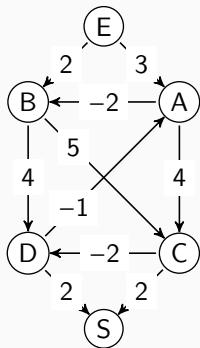
L'algorithme va afficher une erreur car le cycle A-B-C-D-A est de poids total -1 .

Exemple Bellman-Ford



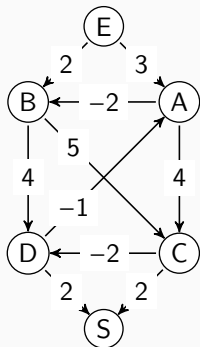
E	A	B	C	D	S	Passage n°
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	0

Exemple Bellman-Ford



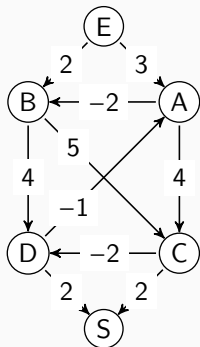
E	A	B	C	D	S	Passage n°
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	0
	3(E)	2(E)	$+\infty$	$+\infty$	$+\infty$	1

Exemple Bellman-Ford



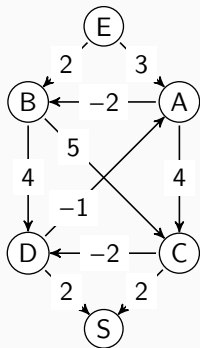
E	A	B	C	D	S	Passage n°
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	0
	3(E)	2(E)	$+\infty$	$+\infty$	$+\infty$	1
	3(E)	1(A)	7(B)	6(B)	$+\infty$	2

Exemple Bellman-Ford



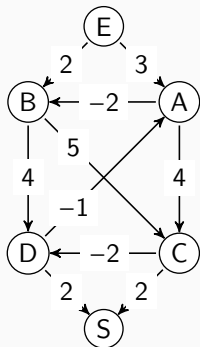
E	A	B	C	D	S	Passage n°
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	0
	3(E)	2(E)	$+\infty$	$+\infty$	$+\infty$	1
	3(E)	1(A)	7(B)	6(B)	$+\infty$	2
	3(E)	1(A)	6(B)	5(C)	8(D)	3

Exemple Bellman-Ford



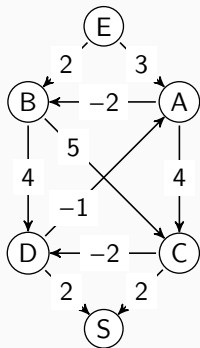
E	A	B	C	D	S	Passage n°
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	0
	3(E)	2(E)	$+\infty$	$+\infty$	$+\infty$	1
	3(E)	1(A)	7(B)	6(B)	$+\infty$	2
	3(E)	1(A)	6(B)	5(C)	8(D)	3
	3(E)	1(A)	6(B)	4(C)	7(D)	4

Exemple Bellman-Ford



E	A	B	C	D	S	Passage n°
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	0
	3(E)	2(E)	$+\infty$	$+\infty$	$+\infty$	1
	3(E)	1(A)	7(B)	6(B)	$+\infty$	2
	3(E)	1(A)	6(B)	5(C)	8(D)	3
	3(E)	1(A)	6(B)	4(C)	7(D)	4
	3(E)	1(A)	6(B)	4(C)	6(D)	5

Exemple Bellman-Ford



E	A	B	C	D	S	Passage n°
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	0
	3(E)	2(E)	$+\infty$	$+\infty$	$+\infty$	1
	3(E)	1(A)	7(B)	6(B)	$+\infty$	2
	3(E)	1(A)	6(B)	5(C)	8(D)	3
	3(E)	1(A)	6(B)	4(C)	7(D)	4
	3(E)	1(A)	6(B)	4(C)	6(D)	5
	3(E)	1(A)	6(B)	4(C)	6(D)	vérification

On est censé faire 5 passages, mais dès qu'un passage s'est fait sans modification, c'est qu'il n'y aura plus de modification.

La dernière information de chaque colonne donne la distance minimale depuis E ainsi que le sommet d'où l'on vient, ce qui permet de reconstituer le trajet.

Par exemple, le plus court chemin de E vers S est de poids 6 : $E - A - B - C - D - S$.

Pour accélérer l'algorithme, il est possible de regarder les valeurs déjà calculées dans l'itération en cours.

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

Étant donné un graphe valué, on cherche un sous-graphe partiel qui soit un arbre (*arbre recouvrant*) et dont la somme des poids des arêtes soit minimale.

Algorithme de Prim 1957 (Jarník 1930)

Données : un graphe $G = (V, E, p)$ pondéré par une fonction p

Résultat : un arbre recouvrant T (ensemble d'arêtes) de poids minimal val

début

choisir un sommet de départ $x_0 \in V$;

$ATTEINT \leftarrow x_0$;

$T \leftarrow \emptyset$;

$val \leftarrow 0$;

tant que $ATTEINT \neq V$ **faire**

trouver $xy \in E$ de poids minimal avec $x \in ATTEINT$ et

$y \notin ATTEINT$;

$T \leftarrow T + \{xy\}$;

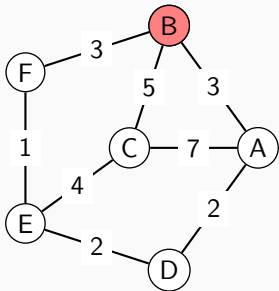
$val \leftarrow val + p(xy)$;

fin

retourner T et val

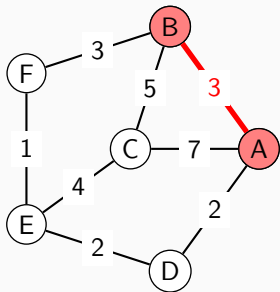
fin

Exemple Prim partant de B



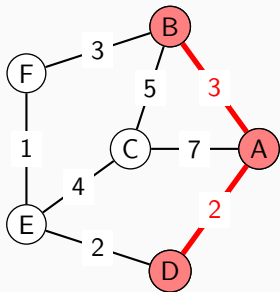
À chaque étape de Prim, notre arbre grandit.

Exemple Prim partant de B



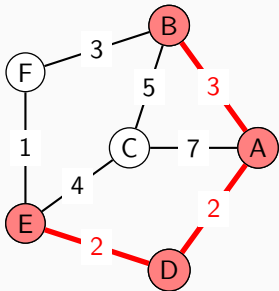
À chaque étape de Prim, notre arbre grandit.

Exemple Prim partant de B



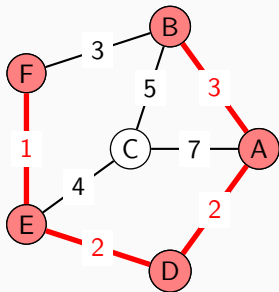
À chaque étape de Prim, notre arbre grandit.

Exemple Prim partant de B



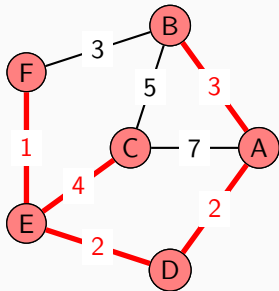
À chaque étape de Prim, notre arbre grandit.

Exemple Prim partant de B



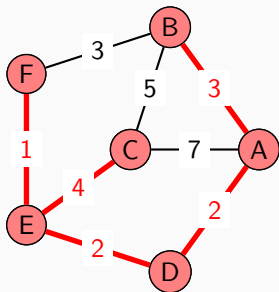
À chaque étape de Prim, notre arbre grandit.

Exemple Prim partant de B



À chaque étape de Prim, notre arbre grandit.

Exemple Prim partant de B



À chaque étape de Prim, notre arbre grandit.
On obtient un ARM de poids 12.

Algorithme de Kruskal 1956

Données : un graphe $G = (V, E, p)$ pondéré par une fonction p

Résultat : un arbre recouvrant T (ensemble d'arêtes) de poids minimal val

début

$L \leftarrow$ liste des arêtes triées par poids croissant ;

$T \leftarrow \emptyset$;

$val \leftarrow 0$;

pour $e \in L$ **faire**

si $T + \{e\}$ *n'a pas de cycle* **alors**

$T \leftarrow T + \{e\}$;

$val \leftarrow val + p(e)$;

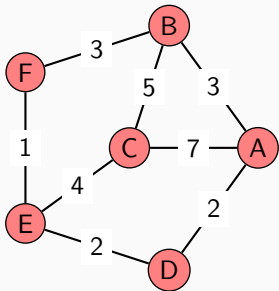
fin

fin

retourner T et val

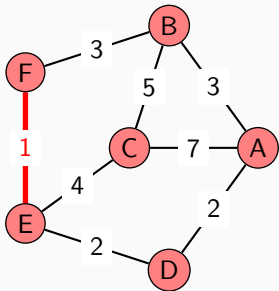
fin

Exemple Kruskal



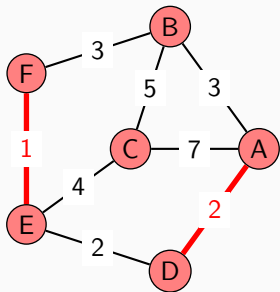
À chaque étape de Kruskal, deux arbres s'unissent.

Exemple Kruskal



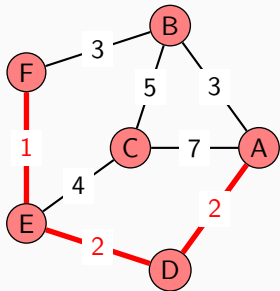
À chaque étape de Kruskal, deux arbres s'unissent.

Exemple Kruskal



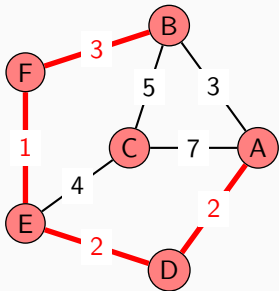
À chaque étape de Kruskal, deux arbres s'unissent.

Exemple Kruskal



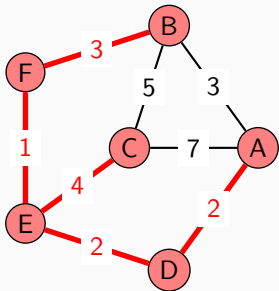
À chaque étape de Kruskal, deux arbres s'unissent.

Exemple Kruskal



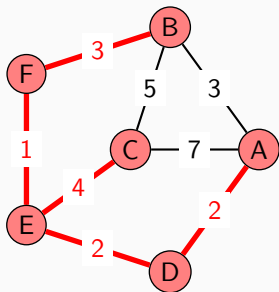
À chaque étape de Kruskal, deux arbres s'unissent.

Exemple Kruskal



À chaque étape de Kruskal, deux arbres s'unissent.

Exemple Kruskal



À chaque étape de Kruskal, deux arbres s'unissent.
On obtient un ARM de poids 12.

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnancement

Définitions

- Un *réseau de flot* est un graphe simple valué $G = (V, E, c)$;

Définitions

- Un *réseau de flot* est un graphe simple valué $G = (V, E, c)$;
- $c : E \rightarrow \mathbb{R}^+$ est la *capacité* du réseau ;

Définitions

- Un *réseau de flot* est un graphe simple valué $G = (V, E, c)$;
- $c : E \rightarrow \mathbb{R}^+$ est la *capacité* du réseau ;
- $s \in V$ est la *source*, $p \in V$ est le *puits*.

Définitions

- Un *réseau de flot* est un graphe simple valué $G = (V, E, c)$;
- $c : E \rightarrow \mathbb{R}^+$ est la *capacité* du réseau ;
- $s \in V$ est la *source*, $p \in V$ est le *puits*.
- Un *flot* est une fonction $f : E \rightarrow \mathbb{R}^+$ qui vérifie deux conditions :

Définitions

- Un *réseau de flot* est un graphe simple valué $G = (V, E, c)$;
- $c : E \rightarrow \mathbb{R}^+$ est la *capacité* du réseau ;
- $s \in V$ est la *source*, $p \in V$ est le *puits*.
- Un *flot* est une fonction $f : E \rightarrow \mathbb{R}^+$ qui vérifie deux conditions :
 - $\text{flot} \leq \text{capacité}$:

$$\forall e \in E \quad f(e) \leq c(e)$$

Définitions

- Un *réseau de flot* est un graphe simple valué $G = (V, E, c)$;
- $c : E \rightarrow \mathbb{R}^+$ est la *capacité* du réseau ;
- $s \in V$ est la *source*, $p \in V$ est le *puits*.
- Un *flot* est une fonction $f : E \rightarrow \mathbb{R}^+$ qui vérifie deux conditions :

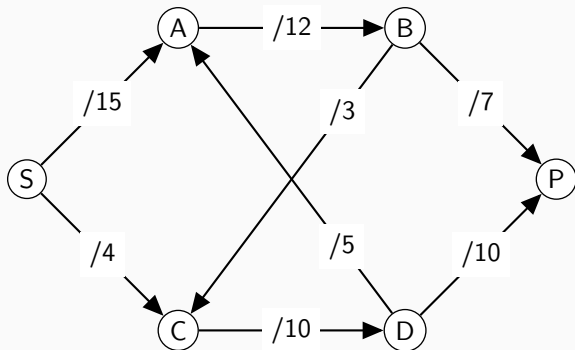
- $f \leq c$:

$$\forall e \in E \quad f(e) \leq c(e)$$

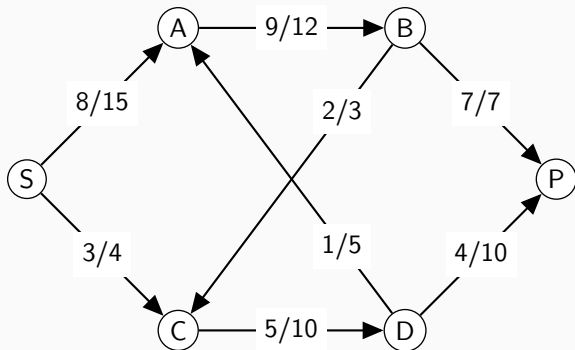
- $\text{flot entrant} = \text{flot sortant}$ (*conservation du flot*) :

$$\forall v \in V \setminus \{s, p\} \quad \sum_{vy \in E} f(vy) = \sum_{xv \in E} f(xv)$$

Exemple



Exemple



un flot possible de valeur 11

- Par conservation du flot en chaque nœud, le flot arrivant en P est égal au flot sortant de S .

- Par conservation du flot en chaque nœud, le flot arrivant en P est égal au flot sortant de S .
- Comment rendre ce flot maximal ?

- Par conservation du flot en chaque nœud, le flot arrivant en P est égal au flot sortant de S .
- Comment rendre ce flot maximal ?
- On va pouvoir parcourir les arcs éventuellement en sens inverse.

- Par conservation du flot en chaque nœud, le flot arrivant en P est égal au flot sortant de S .
- Comment rendre ce flot maximal ?
- On va pouvoir parcourir les arcs éventuellement en sens inverse.

Définition

Soit $(x, y) \in E$.

- L'arc (x, y) (sens direct) est dit *saturé* si $f(x, y) = c(x, y)$.
- L'arc (y, x) (sens inverse) est dit *saturé* si $f(x, y) = 0$.

- Par conservation du flot en chaque nœud, le flot arrivant en P est égal au flot sortant de S .
- Comment rendre ce flot maximal ?
- On va pouvoir parcourir les arcs éventuellement en sens inverse.

Définition

Soit $(x, y) \in E$.

- L'arc (x, y) (sens direct) est dit *saturé* si $f(x, y) = c(x, y)$.
- L'arc (y, x) (sens inverse) est dit *saturé* si $f(x, y) = 0$.

Définition

Une chaîne (suite d'arcs peu importe leur orientation) est dite *améliorante* si elle est constituée d'arcs non saturés.

Notons E^+ (resp. E^-) l'ensemble des arcs de sens direct (resp. indirect) d'une chaîne améliorante. On pose

$$\varepsilon^+ = \min_{e \in E^+} c(e) - f(e)$$

$$\varepsilon^- = \min_{e \in E^-} f(e)$$

$$\varepsilon = \min(\varepsilon^+, \varepsilon^-)$$

On peut alors augmenter le flot de ε :

- chaque arc de E^+ voit son flot augmenté de ε ;
- chaque arc de E^- voit son flot diminué de ε .

Algorithme de Ford-Fulkerson 1962

Données : un réseau de flot $G = (V, E, c)$ de capacité c

Résultat : un flot maximal

début

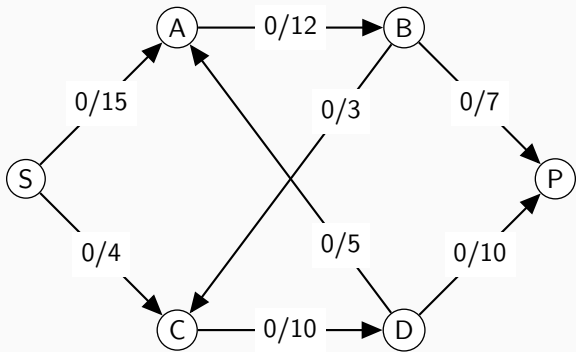
tant que *il existe une chaîne améliorante* **faire**

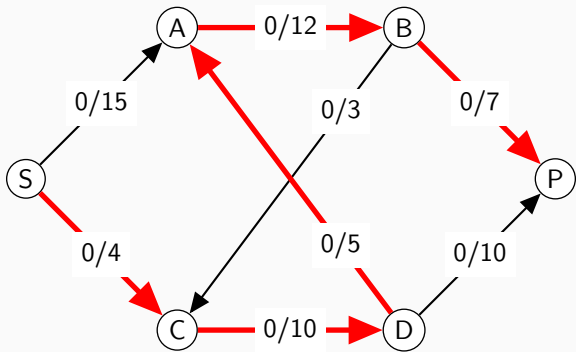
 | Améliorer le flot ;

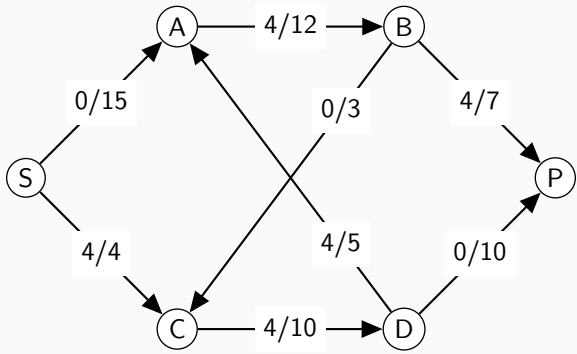
fin

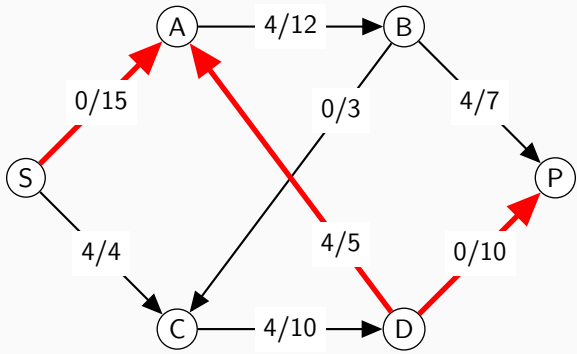
 retourner *flot*

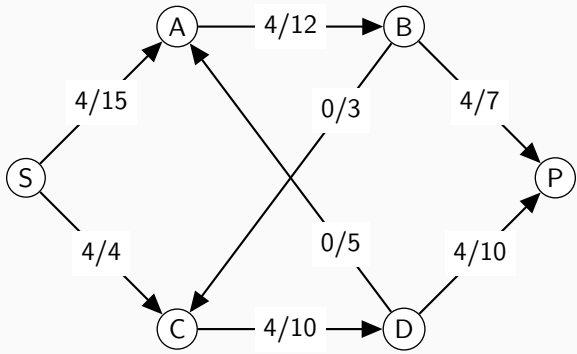
fin

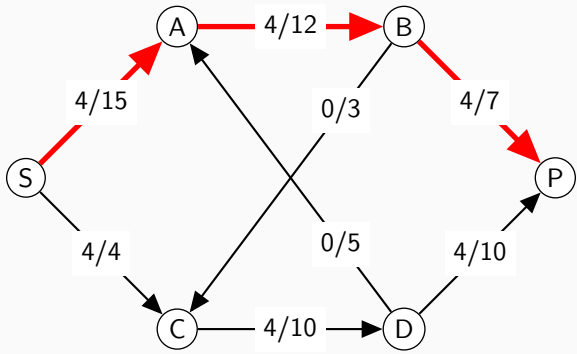


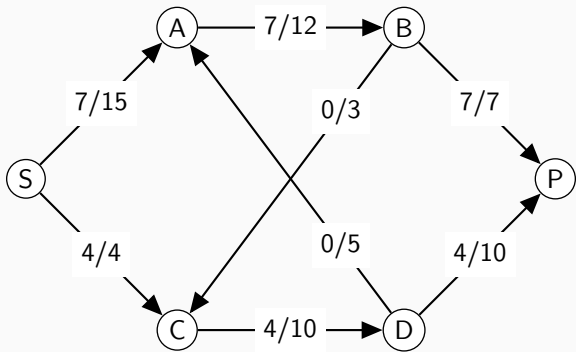


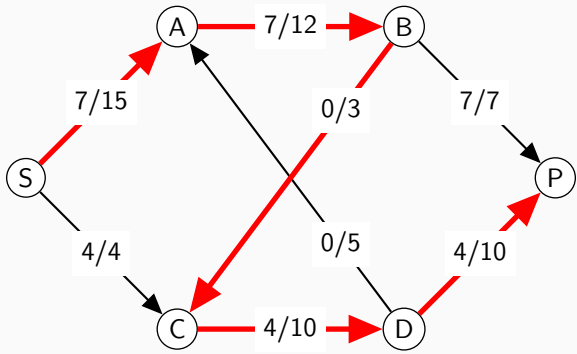


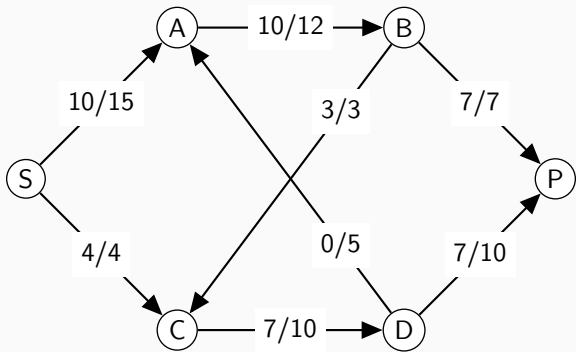


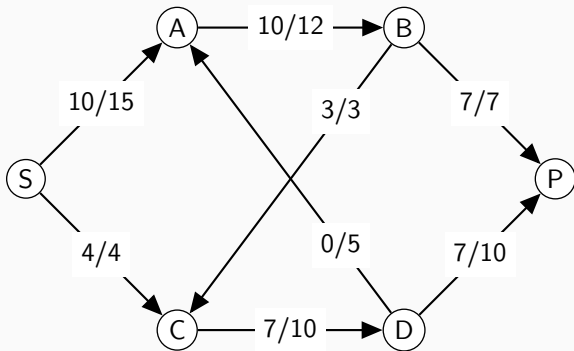












flot maximal de valeur 14

Peut-on anticiper la valeur du flot maximal ?

Peut-on anticiper la valeur du flot maximal ?

Définitions

- Une *coupe* est une partition de V de la forme (X, \bar{X}) avec $s \in X$ et $p \in \bar{X}$.
- La *capacité* d'une coupe est

$$\sum_{\substack{xy \in E \\ x \in X \\ y \in \bar{X}}} c(x, y).$$

Peut-on anticiper la valeur du flot maximal ?

Définitions

- Une *coupe* est une partition de V de la forme (X, \overline{X}) avec $s \in X$ et $p \in \overline{X}$.
- La *capacité* d'une coupe est

$$\sum_{\substack{xy \in E \\ x \in X \\ y \in \overline{X}}} c(x, y).$$

- Une coupe est *minimale* si sa capacité est minimale parmi toutes les coupes possibles.

Théorème flot maximal/coupe minimale (1956)

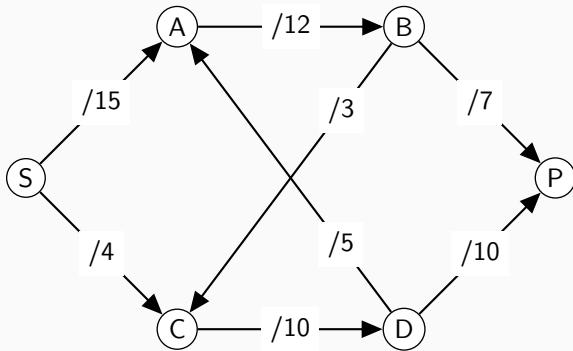
Théorème

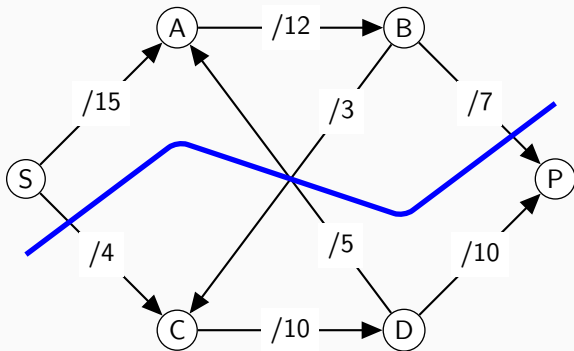
La valeur d'un flot maximal est égal à la valeur d'une coupe minimale.

De plus toutes les arcs de la coupe minimale (ayant donc leur origine dans X et leur extrémité dans \bar{X}) sont saturés par le flot maximal (« goulot d'étranglement »).

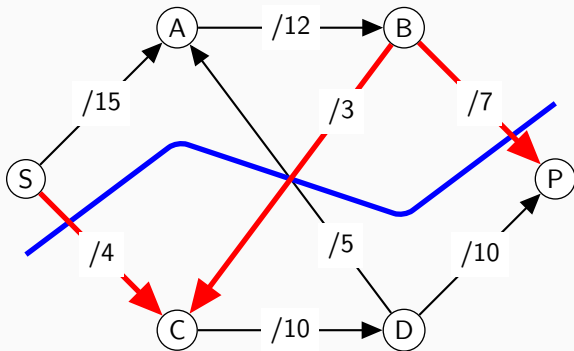
mpechaud.fr/scripts/maxflow/index.html

www.cambridge.org/core/services/aop-cambridge-core/content/view/5D6E55D3B06C4F7B1043BC1D82D40764/S0008414X00036890a.pdf/maximal_flow_through_a_network.pdf





$X = \{S, A, B\}$ et $\bar{X} = \{C, D, P\}$



$X = \{S, A, B\}$ et $\bar{X} = \{C, D, P\}$

est une coupe minimale

1. Généralités

Définitions

Représentations

Familles de graphes

Sous-graphes

Isomorphismes

2. Graphes et chemins

Chaînes et cycles

Connexité

Graphes eulériens / hamiltoniens

Parcours

3. Graphes et couleurs

Nombre chromatique

Planarité et coloration

4. Graphes valués

Plus court chemin

Arbre recouvrant minimal

Graphes de flot

Ordonnement

Exemple

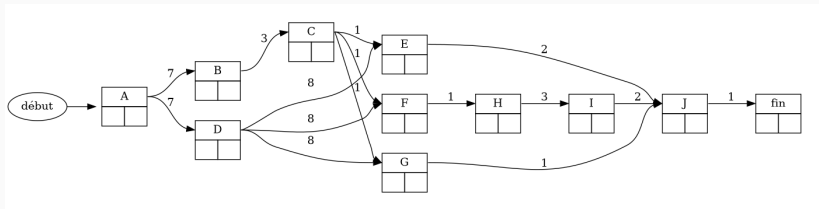
La réalisation d'un projet nécessite un certain nombre de tâches dont les durées et les contraintes d'antériorité sont les suivantes :

Tâches	Durées	Tâches antérieures
A	7	-
B	3	A
C	1	B
D	8	A
E	2	D, C
F	1	D, C
G	1	D, C
H	3	F
I	2	H
J	1	E, G, I

Représentation MPM

Pour la représentation du graphe MPM associé à la réalisation de ce projet, il est fortement conseillé de tester que le graphe est sans circuit ;

Niveau	0	1	2	1	3	3	3	4	5	6
Sommets	A	B	C	D	E	F	G	H	I	J
Précédents		A	B	A	D, C	D, C	D, C	F	H	E, G, I



Si le projet commence à la date $t = 0$, la durée totale du projet est égale à la longueur d'un plus long chemin de début à fin.

Dates

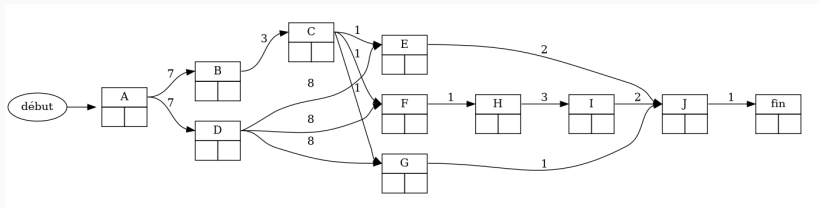
- T_i (début de la tâche i) est la date à laquelle peut commencer au plus tôt la tâche i .
- Il s'agit de la longueur d'un plus long chemin, au sens des délais dans le graphe, reliant le sommet 1 au sommet i .

Les dates au plus tôt se calculent en partant du sommet initial :

$$T_1 = 0$$

$$T_i = \max_{p \in P(i)} [T_p + d_{p,i}]$$

P désigne les précédents.



Dates

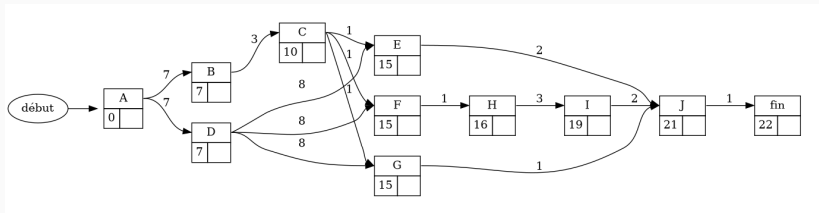
- T_i (début de la tâche i) est la date à laquelle peut commencer au plus tôt la tâche i .
- Il s'agit de la longueur d'un plus long chemin, au sens des délais dans le graphe, reliant le sommet 1 au sommet i .

Les dates au plus tôt se calculent en partant du sommet initial :

$$T_1 = 0$$

$$T_i = \max_{p \in P(i)} [T_p + d_{p,i}]$$

P désigne les précédents.

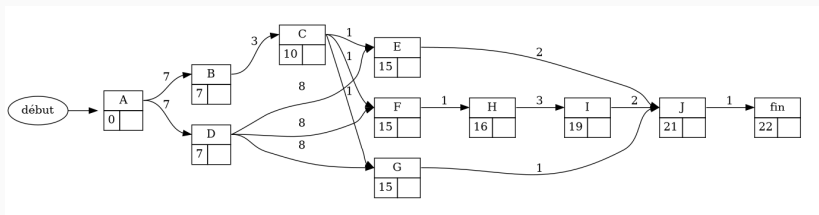


Dates

Il s'agit, pour une tâche i , de connaître le retard de sa mise en route que l'on peut se permettre sans pour cela modifier la date de fin des travaux. Les tâches critiques sont celles que l'on ne peut pas retarder. Pour les autres, un retard sans retard final. On les calcule cette fois-ci en partant de la fin. En notant T^* les dates au plus tard

$$T_n^* = T_n$$
$$T_i^* = \min_{s \in S(i)} [T_s^* - d_{i,s}]$$

S désigne les successeurs.

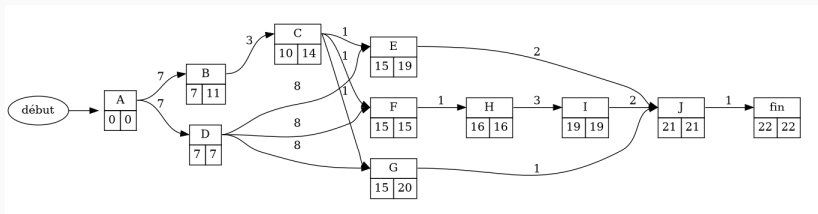


Dates

Il s'agit, pour une tâche i , de connaître le retard de sa mise en route que l'on peut se permettre sans pour cela modifier la date de fin des travaux. Les tâches critiques sont celles que l'on ne peut pas retarder. Pour les autres, un retard sans retard final. On les calcule cette fois-ci en partant de la fin. En notant T^* les dates au plus tard

$$T_n^* = T_n$$
$$T_i^* = \min_{s \in S(i)} [T_s^* - d_{i,s}]$$

S désigne les successeurs.



Mathématiciens, par ordre de citation

Leonhard Euler (1707-1783) : suisse

William Rowan Hamilton (1805-1865) : irlandais

Øystein Ore (1899-1968) : norvégien

Gabriel Andrew Dirac (1925-1984) : hongro-anglais (né en Hongrie, mort en Suisse, a enseigné au Danemark)

Dominic James Anthony Welsh (1938-) : anglais

Martin Beynon Powell (XX^e) : anglais

Kazimierz Kuratowski (1896-1980) : polonais

Percy John Heawood (1861-1955) : britannique

Kenneth Ira Appel (1932-2013) : américain

Wolfgang Haken (1928-) : allemand

Edsger Wybe Dijkstra (1930-2002) : néerlandais

Richard Ernest Bellman (1920-1984) : américain

Lester Randolph Ford Jr. (1927-2017) : américain

Robert Clay Prim (1921-) : américain

Joseph Bernard Kruskal Jr. (1928-2010) : américain

Delbert Ray Fulkerson (1924-1976) : américain