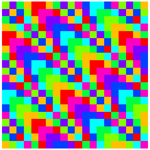
Introduction aux mathématiques discrètes (2)

Michel Rigo

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

Année 2018–2019



- Cours 1 : graphes, chemins, connexité
- Aujourd'hui :
  - sous-graphes
  - arbres
  - chemin de poids minimal
  - planarité

Sous-graphe

Définition

Soit $G = (V, E)$ un graphe (orienté ou non).
Le graphe $G' = (V', E')$ est un sous-graphe de G si

- $V' \subseteq V$,
- $E' \subseteq E \cap (V' \times V')$.

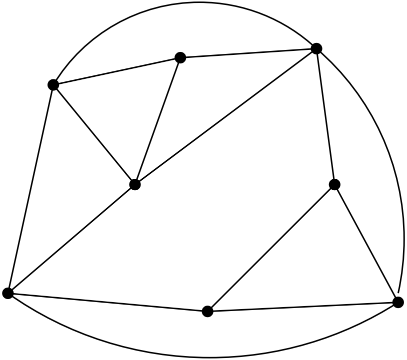
⚠ Si on enlève un sommet v de G,
il faut enlever tous les arcs incidents à v.

Définition

G' est un sous-graphe propre de G si $E' \subsetneq E$ ou $V' \subsetneq V$.

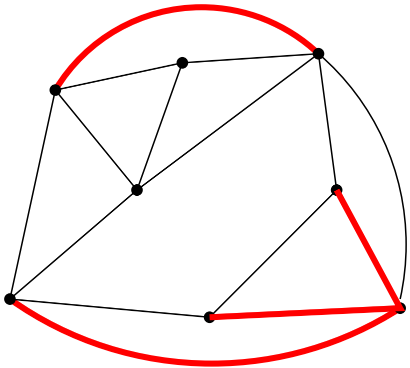
Sous-graphe

suppression d'arêtes



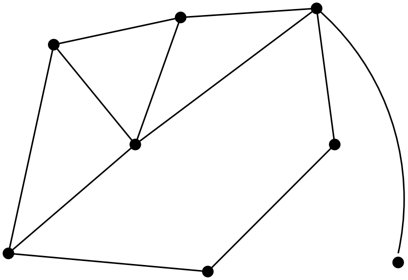
Sous-graphe

suppression d'arêtes



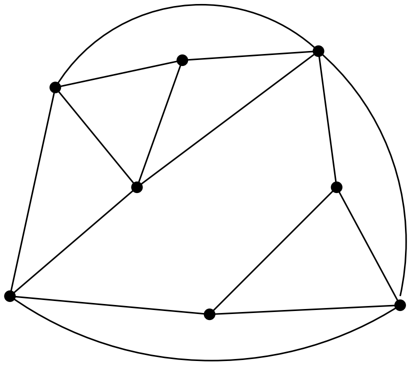
Sous-graphe

suppression d'arêtes

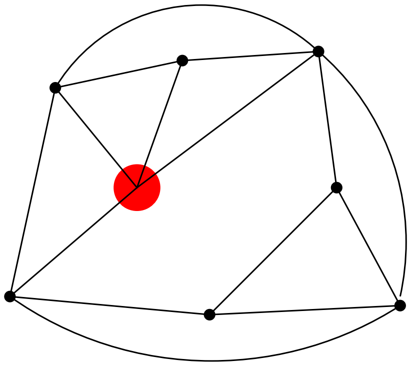


Sous-graphe

suppression de sommets (et arêtes adjacentes)

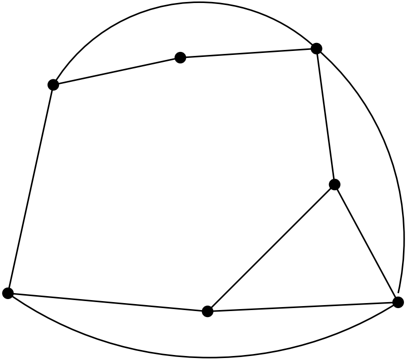


Sous-graphe



Sous-graphe

suppression de sommets (et arêtes adjacentes)



Sous-graphe

Notation

- $G - e$ sous-graphe G' de G obtenu en supprimant l'arc e,
- $G - v$ sous-graphe G' obtenu en supprimant le sommet v et les arcs adjacents,

Extension : si $W = \{v_1, \ldots, v_k\} \subseteq V$, alors $G - W$ est le sous-graphe

$$(\cdots((G - v_1) - v_2) \cdots - v_{k-1}) - v_k := G - v_1 - \cdots - v_k.$$

On procède de même pour un ensemble fini d'arcs et on introduit la notation $G - F$ pour $F \subset E$.

Sous-graphe

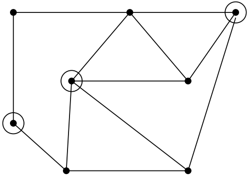
Définition

$W \subseteq V$. Le sous-graphe de G induit par W est
le sous-graphe $G' = (W, E')$ avec $E' = E \cap (W \times W)$.

Définition

Si $W \subseteq V$ est tel que le sous-graphe induit par W ne contient aucune arête, alors les sommets de W sont dits indépendants.

$\alpha(G)$ = nombre maximal de sommets indépendants de G



Sous-graphe

Définition

Soient $G = (V, E)$ un graphe et $G' = (V', E')$ un sous-graphe.

G' est un sous-graphe couvrant G, si $V' = V$ et si

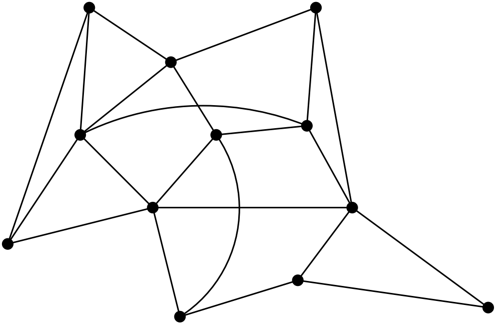
$$\forall v \in V, \exists z \in V : \{z, v\} \in E'.$$

On dira que E' est une couverture (par des arêtes) de G
i.e., tout sommet de G est une extrémité d'une arête de E'.

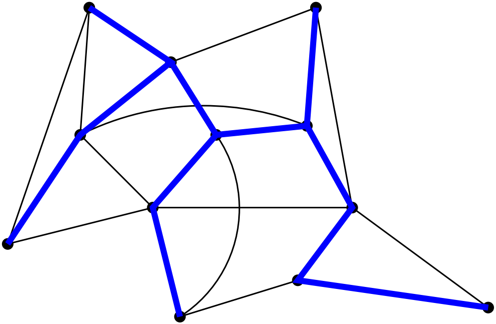
Notion duale : $W \subset V$ est une couverture (par des sommets) si toute arête a une extrémité dans W.

Sous-graphe

sous-graphe couvrant, couverture (par les arêtes)

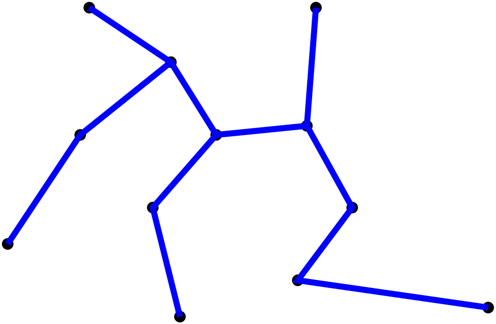


sous-graphe couvrant, couverture (par les arêtes)



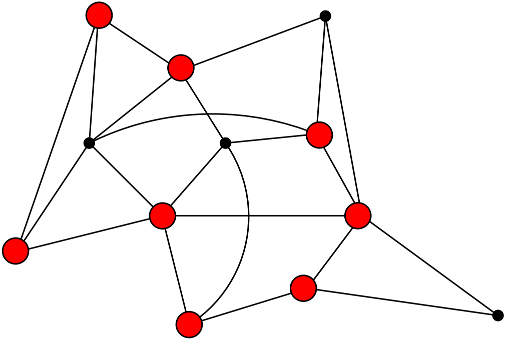
Sous-graphe

sous-graphe couvrant, couverture (par les arêtes)



Sous-graphe

ensemble de domination, couverture (par les sommets)

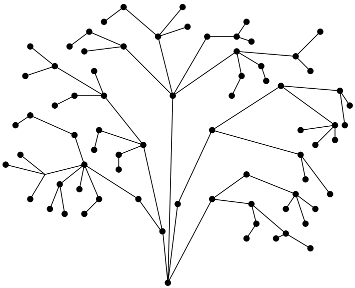


Arbres

Définition

Un arbre est un graphe simple non orienté connexe et sans cycle (i.e., sans circuit simple)

Une forêt est un graphe simple non orienté dont chaque composante connexe est un arbre.



Arbres

Dans un arbre :

- Toute paire de sommets distincts de G est connectée par exactement un chemin simple.
- Les arêtes d'un arbre sont toutes des arêtes de coupure.
- Soit $e \in (V \times V) \setminus E$ qui n'est pas une boucle. Le graphe $G + e$ contient une piste fermée, c'est-à-dire, $G + e$ n'est plus un arbre.

Réciproquement, un graphe connexe dont toutes les arêtes sont des arêtes de coupure est un arbre.

Lemme

Tout arbre non trivial (non réduit à un sommet) contient un sommet de degré 1.

$\rightsquigarrow$ Si un graphe connexe est tel que tout sommet est de degré ≥ 2, alors ce graphe contient un cycle.

Lemme

Si $A = (G, V)$ est un arbre, alors $\#V = \#E + 1$.

Preuve par récurrence sur $\#V$.
OK pour $\#V = 1$ (graphe trivial) ou $\#V = 2$ (K_2).

Si OK pour n, est-ce OK pour $n + 1$?
A possède un sommet v de degré 1, le graphe $A - v$ est encore un arbre et on applique l'hypothèse de récurrence.

Arbres

Proposition

Un graphe est connexe SSI il possède un sous-arbre couvrant.

$\Leftarrow$ Clair.

$\Rightarrow$ Soient $G = (V, E)$ un graphe connexe et $C = (V, E')$ un sous-graphe couvrant *connexe* minimal (i.e., on ne peut pas remplacer E' par un sous-ensemble strict et garder la connexité).

Vu la minimalité de C, chacune de ses arêtes est une *arête de coupure* de C. $\rightsquigarrow$ C est un arbre.

Corollaire

Si $G = (V, E)$ est un graphe (simple non orienté) connexe, alors $\#E \geq \#V - 1$.

G possède un sous-arbre couvrant $C = (V, E')$. De là, il vient

$$\#E \geq \#E' = \#V - 1$$

SpanningTree(G)

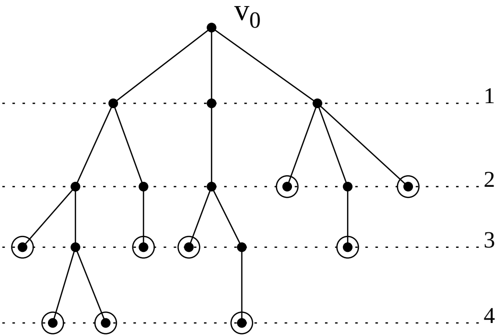
Choose a vertex v ;
$Component \leftarrow \{v\}$; $New \leftarrow \{v\}$; $Edges \leftarrow \emptyset$;
while $New \neq \emptyset$,
    do $Neighbors \leftarrow \emptyset$;
        for all $u \in New$,
            do $Neighbors \leftarrow Neighbors \cup \nu(u)$;
        $New \leftarrow Neighbors \setminus Component$;
        for all $v \in New$,
            do choose an edge $\{u, v\}$ with $u \in Component$;
                $Edges \leftarrow Edges \cup \{\{u, v\}\}$;
        $Component \leftarrow Component \cup New$;
if $Component \neq V(G)$
    then return 'Error : G is not connected' ;
    else **return** $(Component, Edges)$;

Arbres

Définition

Un arbre $A = (V, E)$ avec un sommet privilégié v_0 est un arbre pointé : (A, v_0) et v_0 est la racine de l'arbre.

sommets de A ordonnés selon leur distance à v_0.
Si $d(v_0, v) = i$ v est un sommet de niveau i.



Définition

Si v est un sommet de niveau i et si tous ses voisins sont de niveau $i-1$, on dit alors que v est une feuille de l'arbre.

La hauteur d'un arbre est le niveau maximal de ses feuilles.

Définition

Pointer un arbre définit *une orientation* des arêtes :
orienter les arcs de façon à ce qu'ils joignent les sommets de niveau i aux sommets de niveau $i+1$.

fils (resp. père) d'un noeud v : ses successeurs
(resp. son unique prédécesseur).

descendants (resp. ancêtres) de v : les éléments de $\mathrm{succ}^*(v)$
(resp. $\mathrm{pred}^*(v)$).

DÉFINITION

Un arbre pointé est k-aire si tout sommet a au plus k fils.

Si $k = 2$, on parle d'arbre binaire.

Un arbre k-aire de hauteur n possède au plus

$$1 + k + \cdots + k^n = \frac{k^n - 1}{k - 1}$$

sommets. S'il en possède exactement ce nombre, on parle d'arbre k-aire complet.

Parcours d'arbres

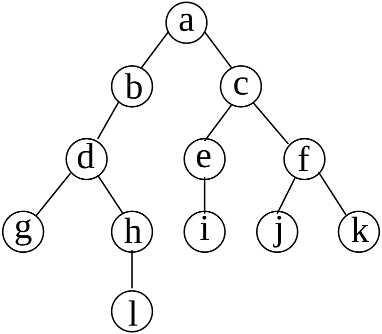
ordonner les noeuds : on suppose que les fils d'un noeud v_i sont ordonnés $v_{i,1}, \ldots, v_{i,k_i}$.
Cet ordre est connu et fixé une fois pour toutes.

Parcours en profondeur

- parcours préfixe : d'abord la racine puis, de manière récursive, les sous-arbres pointés de racine respective $v_{0,1}, \ldots, v_{0,k_0}$.
- parcours suffixe : d'abord, de manière récursive, les sous-arbres pointés de racine $v_{0,1}, \ldots, v_{0,k_0}$, puis la racine v_0.
- parcours infixe (arbre binaire) : d'abord, de manière récursive, le sous-arbre de gauche, puis la racine, et enfin le sous-arbre de droit (Si un sommet n'a qu'un seul fils : sous-arbre de droite vide).

parcours préfixe

d'abord la racine puis, de manière récursive, les sous-arbres pointés

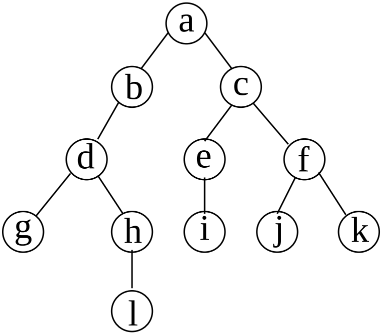


.

Parcours d'arbres

parcours préfixe

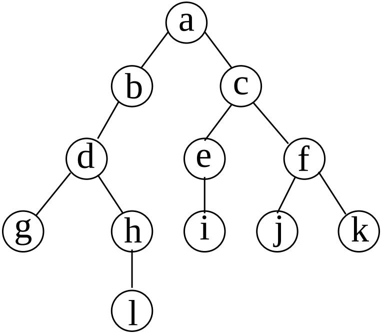
d'abord la racine puis, de manière récursive, les sous-arbres pointés



$a, b, d, g, h, l, c, e, i, f, j, k$

Parcours suffixe

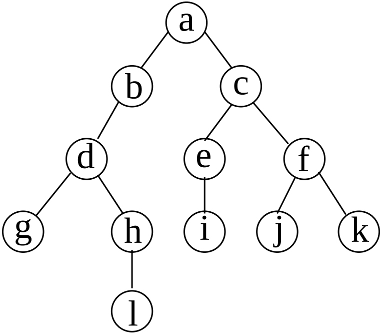
d'abord, de manière récursive, les sous-arbres pointés de racine $v_{0,1}, \ldots, v_{0,k_0}$, puis la racine v_0.



.

Parcours suffixe

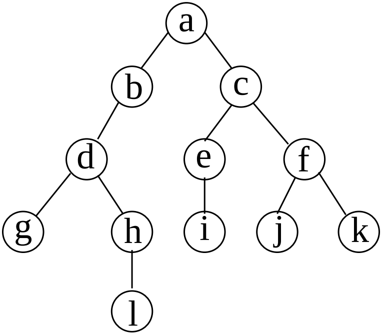
d'abord, de manière récursive, les sous-arbres pointés de racine $v_{0,1}, \ldots, v_{0,k_0}$, puis la racine v_0.



$g, l, h, d, b, i, e, j, k, f, c, a$

Parcours infixe

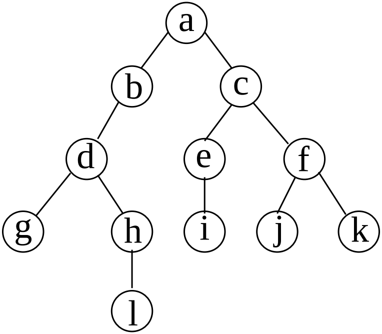
d'abord, de manière récursive, le sous-arbre de gauche, puis la racine, et enfin le sous-arbre de droit (Si un sommet n'a qu'un seul fils : sous-arbre de droite vide).



.

PARCOURS INFIXE

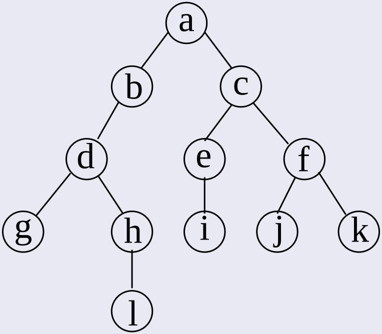
d'abord, de manière récursive, le sous-arbre de gauche, puis la racine, et enfin le sous-arbre de droit (Si un sommet n'a qu'un seul fils : sous-arbre de droite vide).



$g, d, l, h, b, a, i, e, c, j, f, k$

Parcours en largeur

parcours en largeur : parcours des noeuds de l'arbre pointé par niveau croissant.

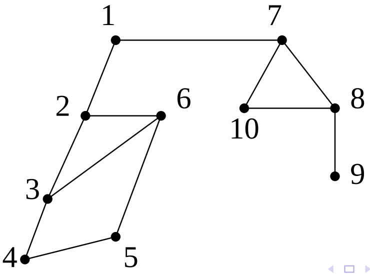


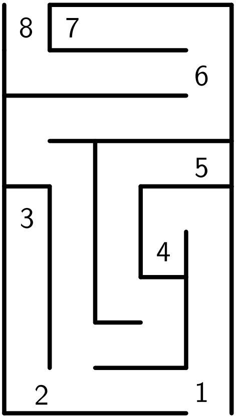
$a, b, c, \ldots, k, l.$

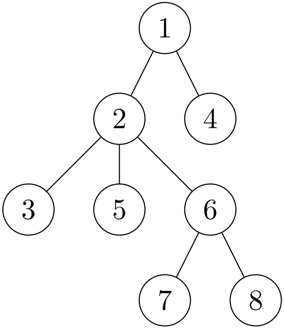
Remarque

Soit $G=(V,E)$ un graphe (orienté ou non) simple et connexe.
Un parcours en profondeur de G est défini récursivement.

Sélectionner un sommet v_0.
A l'étape $k \geq 1$, choisir un voisin de v_{k-1} qui n'a pas encore été sélectionné.
Si un tel voisin n'existe pas, on cherche dans l'ordre, un voisin non sélectionné de $v_{k-2}, \ldots, v_0$.




8
7
6
5
3
4
2
1


1
2
4
3
5
6
7
8

Recherche du plus court chemin

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

un plus court chemin = chemin de poids minimal

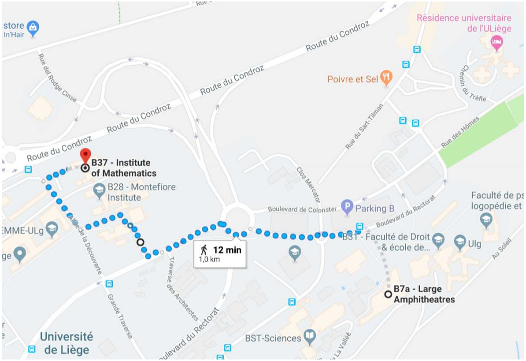
⚠ se restreindre à un graphe orienté simple

Remarque

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

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

Algorithme de Dijkstra



Edsger W. Dijkstra (né en 1930)

Algorithme de Dijkstra

Intuitivement, v_1 est fixé, pour tout sommet v,

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

Lorsque l'algorithme s'achève :

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

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

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

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

Algorithme de Dijkstra

```
Pour tout v ∈ V, T(v):=p(v₁,v), C(v):=(v₁,v)
X:={v₁}
Tant que X≠ V, répéter
   Choisir v ∈ V\X t.q. ∀y ∈ V\X, T(v) ≤T(y)
   X:=X∪{v}
   Pour tout y ∈ V\X
      Si T(y)>T(v) + p(v,y),
         alors T(y):=T(v) + p(v,y) et C(y):=[C(v),y]
```

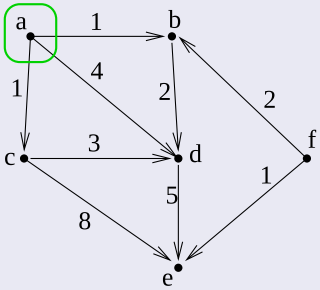
Algorithme de Dijkstra

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

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

Algorithme de Dijkstra

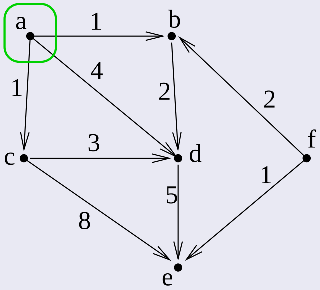
Exemple



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

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

EXEMPLE



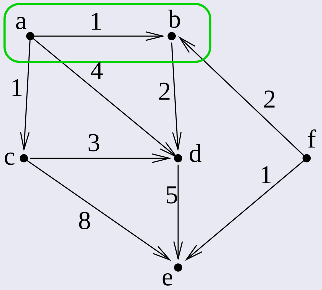
$\mathtt{X} = \{a\}$

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

.

Algorithme de Dijkstra

Exemple

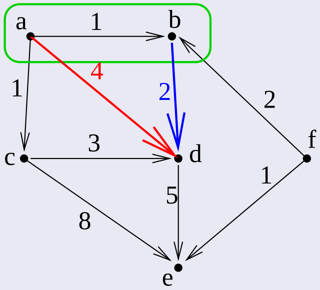


$\mathtt{X} = \{a, b\}$

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

.

Exemple

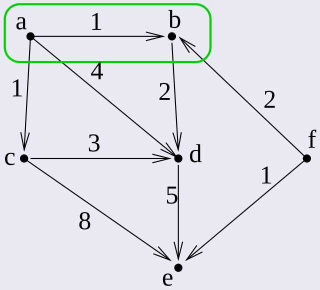


$\mathtt{X} = \{a, b\}$

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

.

Exemple



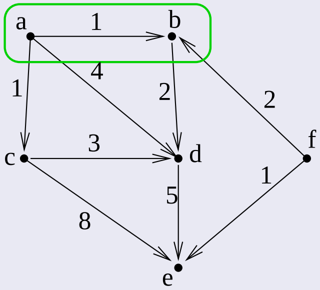
$\mathtt{X} = \{a, b\}$

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

.

Algorithme de Dijkstra

Exemple



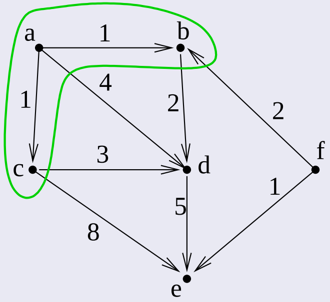
$\mathtt{X} = \{a, b\}$

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

.

Algorithme de Dijkstra

Exemple



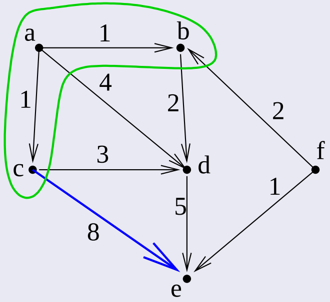
$\mathtt{X} = \{a, b, c\}$

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

.

Algorithme de Dijkstra

Exemple



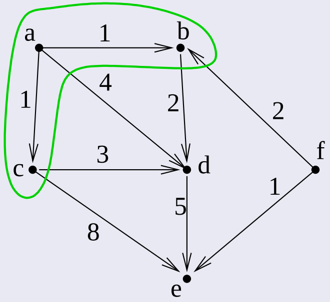
$\mathtt{X} = \{a, b, c\}$

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

.

Algorithme de Dijkstra

Exemple



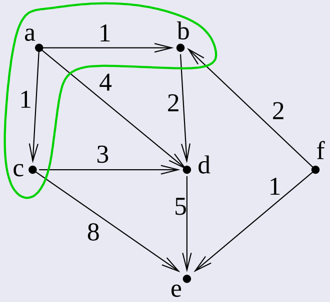
$\mathtt{X} = \{a, b, c\}$

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

.

Algorithme de Dijkstra

Exemple

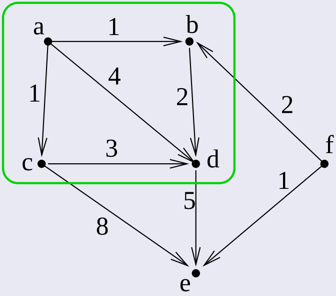


$\mathtt{X} = \{a, b, c\}$

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

.

EXEMPLE



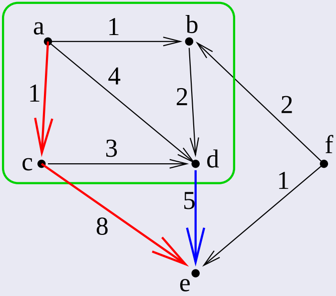
$\mathtt{X} = \{a, b, c, d\}$

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

.

Algorithme de Dijkstra

Exemple

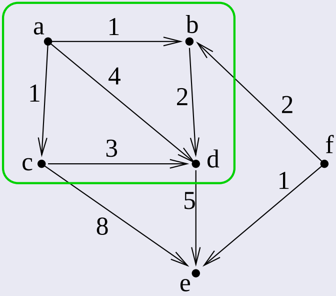


$\mathtt{X} = \{a, b, c, d\}$

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

.

Exemple



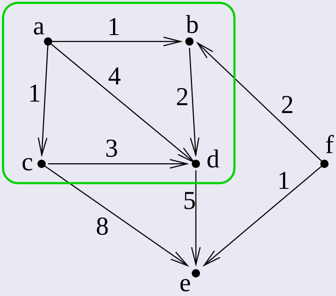
$\mathtt{X} = \{a, b, c, d\}$

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

.

Algorithme de Dijkstra

Exemple

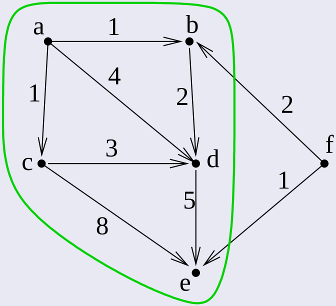


$\mathtt{X} = \{a, b, c, d\}$

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

.

Exemple



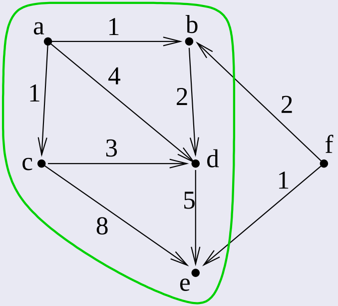
$\mathtt{X} = \{a, b, c, d, e\}$

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

.

Algorithme de Dijkstra

Exemple

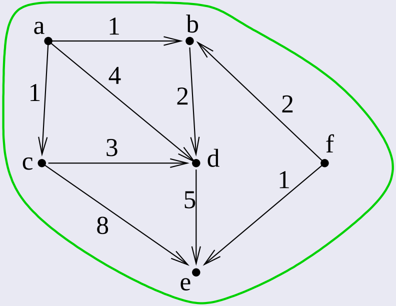


$\mathtt{X} = \{a, b, c, d, e\}$

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

.

Exemple



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

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

.

Preuve de l'algorithme (évocation)

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

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

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

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

$$T_{j+1}(y) \leq T_j(y)$$

Preuve de l'algorithme (évocation)

Récurrence sur j, on va montrer que

I) $\forall v \in X_j$, $T_j(v)$ est le poids minimal de *tous* les chemins joignant v_1 à v.

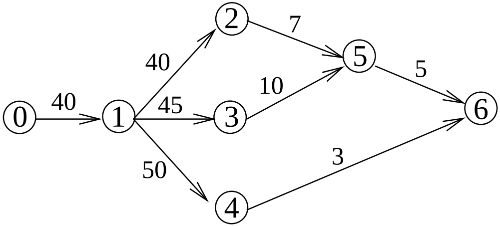
II) $\forall v \notin X_j$, $T_j(v)$ est le poids minimal des chemins joignant v_1 à v qui, à l'exception de v, passent uniquement par des sommets de X_j.

D'où le résultat pour $j = \# V$.

Chemin critique

Chemin critique

- A : Creusage des fondations
- B : Construction du gros-oeuvre
- C : Installation électrique
- D : Installation du chauffage central
- E : Réalisation des peintures extérieures
- F : Réalisation des peintures intérieures



Chemin critique

A : Creusage des fondations

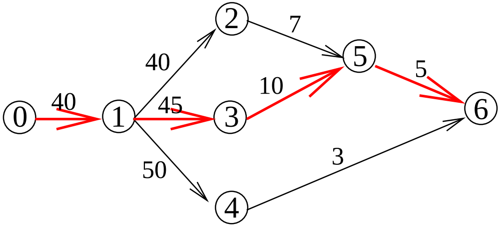
B : Construction du gros-oeuvre

C : Installation électrique

D : Installation du chauffage central

E : Réalisation des peintures extérieures

F : Réalisation des peintures intérieures

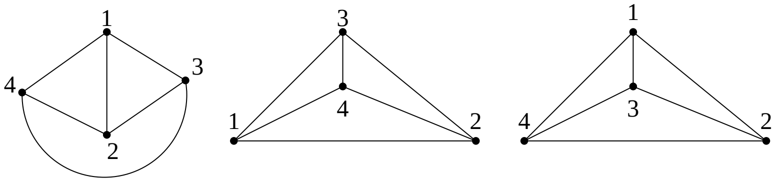


Si on dispose d'un graphe connexe pondérée

$\rightsquigarrow$ recherche d'un sous-arbre couvrant de poids minimum

- Algorithme de Prim (même philosophie que Dijkstra)
- Algorithme de Kruskal
- ...

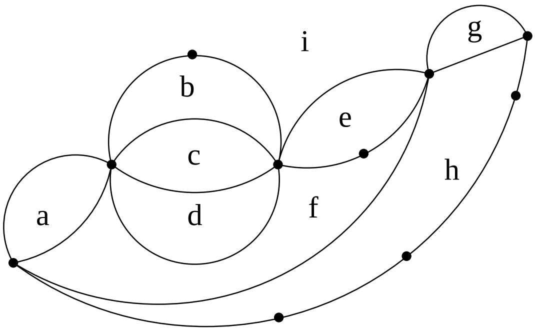
On considère l'ensemble quotient des représentations homéomorphes (au sens topologique du terme).
Un graphe peut avoir plusieurs *représentations* :



Il existe une représentation pour laquelle les arêtes (représentées dans le plan) ne se coupent pas.

Graphes planaires

Face, frontière, faces adjacentes, face infinie...



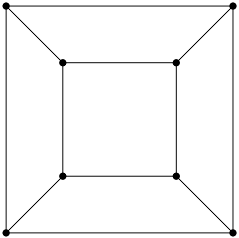
Mathématiquement bien défini : courbe de Jordan $\mathcal{C}$ dans $\mathbb{R}^2$ détermine deux composantes connexes de $\mathbb{R}^2 \setminus \mathcal{C}$.

Le nombre de sommets, d'arêtes sont des invariants,
au vu de la formule d'Euler..., le nombre de face aussi.

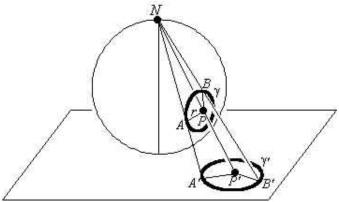
Lien avec la 'géométrie' et l'étude des polyèdres

Remarque : théorème de Steinitz

Un graphe est le squelette d'un polyèdre convexe (borné) de $\mathbb{R}^3$ si et seulement si c'est un graphe planaire au moins 3-connexe (i.e., ne pouvant pas être disconnecté en retirant moins de trois sommets)



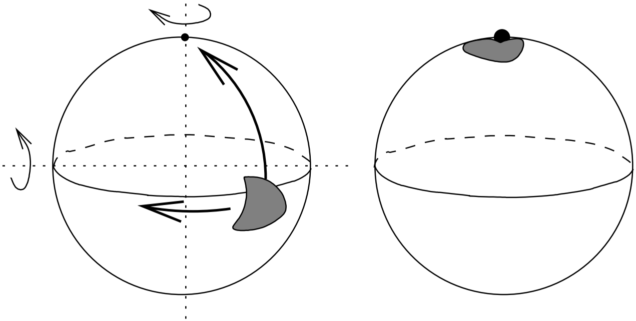
projection stéréographique



https ://math.stackexchange.com/questions/2024598 http ://www.dimensions-math.org

Graphes planaires

projection stéréographique



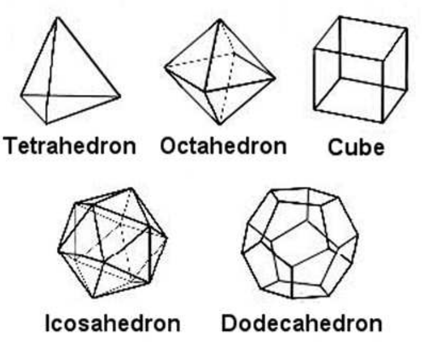
On peut *choisir* quelle face sera la face infinie.

Graphes planaires

Théorème d'Euler

Dans un multi-graphe planaire connexe (fini) possédant s sommets, a arêtes et f faces, on a

$$\boxed{s - a + f = 2.}$$



Récurrence sur f.

Si $f = 1$, le graphe possède uniquement une face infinie.
Le graphe connexe ne possède aucun cycle, c'est un arbre.
Ainsi, $s = a + 1$, formule OK.

OK pour $< f$, OK ? pour $f \geq 2$?.
Soit e une arête d'un cycle.
e appartient à la frontière de deux faces A et B.
Si on supprime e, on obtient un graphe ayant

- $a - 1$ arêtes,
- le même nombre s de sommets,
- $f - 1$ faces.

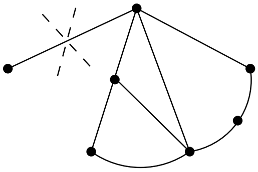
Par hypothèse de récurrence, on a $s - (a - 1) + f - 1 = 2$, ce qui suffit.

Corollaire

Dans un graphe $G = (V, E)$ **simple** et planaire,
il existe un sommet x tel que $\deg(x) \leq 5$.

Quitte à considérer séparément chaque composante connexe de G, on suppose G connexe.

Eliminer les arêtes ne délimitant pas de face (celles-ci ont une extrémité de degré $1 \rightarrow$ résultat trivial).



G est simple, la frontière de toute face contient ≥ 3 arêtes.

En passant en revue les faces (en les compant),

- on compte à chaque fois au moins 3 arêtes
- chaque arête est comptée deux fois
  car elle apparaît dans la frontière de deux faces.

Donc

$$\boxed{3\,f \leq 2\,a.}$$

Par l'absurde. Supposons que pour tout sommet x, $\deg(x) \geq 6$.
Dans ce cas, en passant en revue les sommets, on a

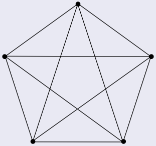
$$6\,s \leq 2\,a.$$

Si on applique la formule d'Euler,

$$2 = s - a + f \leq \frac{a}{3} - a + \frac{2\,a}{3} = 0!$$

Proposition

Le graphe K_5 n'est pas planaire.



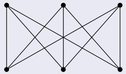
Dans un graphe simple et planaire, de la relation $3f \leq 2a$ démontrée dans la prop. préc. et de la formule d'Euler $(3a - 3f = 3s - 6)$, on tire que

$$a \leq 3s - 6.$$

Or, K_5 est un graphe simple qui possède 5 sommets et 10 arêtes et $10 \not\leq 3.5 - 6$. On en conclut que K_5 ne peut être planaire.

Proposition

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



Ici, graphe simple, planaire et biparti.
Chaque face a une frontière déterminée par au moins 4 arêtes.
On en tire que $4f \leq 2a$, i.e., $2f \leq a$. De la formule d'Euler, $2a - 2f = 2s - 4$, on en tire que

$$a \leq 2s - 4.$$

Or, $K_{3,3}$ est un graphe biparti simple qui possède 6 sommets et 9 arêtes. Il ne peut donc pas être planaire car $9 \not\leq 2.6 - 4$.

Théorème d'exclusion caractérisant une famille de graphes :

Théorème de Kuratowski

Un multi-graphe (non orienté) est planaire si et seulement si il ne contient pas de sous-graphe homéomorphe à K_5 ou à $K_{3,3}$.

Graphes homéomorphes