Chapitre 2
Cheminement dans les Graphes

Présenté par :
H. BENKAOUHA
Bureau 222, Département Informatique, USTHB
hbenkaouha@usthb.dz
haroun.benkaouha@gmail.com

Chaîne

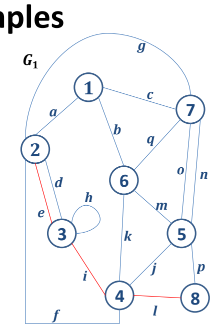
- <u>Chaîne</u> dans un graphe non orienté (resp. orienté) $G=(X, E)$ (resp. $G=(X, U)$):
- $\Rightarrow$Suite alternée de sommets et d'arêtes (resp. d'arcs) :
  - $\mu = x_0\ e_1\ x_1 \ldots x_{k\text{-}1}\ e_k\ x_k$
  - (resp. $\mu = x_0\ u_1\ x_1 \ldots x_{k\text{-}1}\ u_k\ x_k$)
- Tel que pour i de 1 à k,
  - $x_{i\text{-}1}$ et x_i sont extrémités de l'arête e_i (resp. de l'arc u_i).
- On dit que μ est une chaîne joignant les sommets x_0 et x_k de longueur k.

Chaîne - Remarque

- La notion de chaîne est une notion non orientée .
- Mais on peut l'appliquer sur les graphes orientés.
- Il suffit de ne pas prendre en considération le sens des arcs.
- C'est-à-dire, on peut prendre un arc dans le sens inverse.

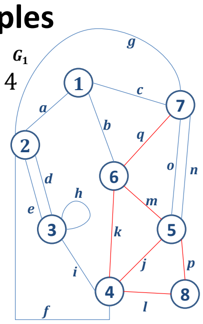
Chaîne - Exemples

- $\mu_1 = 8\ l\ 4\ i\ 3\ e\ 2$
  - Chaîne dans G_1
  - Joignant 8 et 2
  - De longueur 3



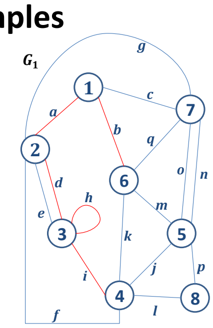
Chaîne - Exemples

- $\mu_2 = 7\ q\ 6\ m\ 5\ j\ 4\ k\ 6\ m\ 5\ p\ 8\ l\ 4$
  - Chaîne dans G_1
  - Joignant 7 et 4
  - De longueur 7



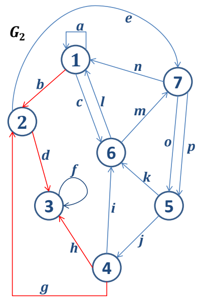
Chaîne - Exemples

- $\mu_3 = 6\ b\ 1\ a\ 2\ d\ 3\ h\ 3\ i\ 4$
  - Chaîne dans G_1
  - Joignant 6 et 4
  - De longueur 5



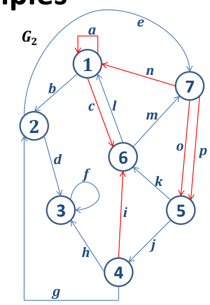
Chaîne - Exemples

- $\mu_4 = 4\ g\ 2\ d\ 3\ h\ 4\ g\ 2\ b\ 1$
  - Chaîne dans G_2
  - Joignant 4 et 1
  - De longueur 5



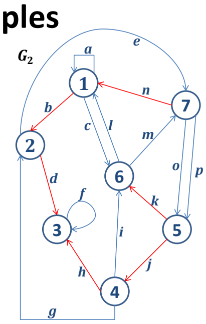
Chaîne - Exemples

- $\mu_5 = 7\ o\ 5\ p\ 7\ n\ 1\ a\ 1\ c\ 6\ i\ 4$
  - Chaîne dans G_2
  - Joignant 7 et 4
  - De longueur 6



Chaîne - Exemples

- $\mu_6 = 6\ k\ 5\ j\ 4\ h\ 3\ d\ 2\ b\ 1\ n\ 7$
  - Chaîne dans G_2
  - Joignant 6 et 7
  - De longueur 6

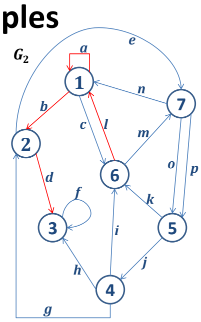


Chemin

- <u>Chemin</u> dans un graphe orienté $G=(X, U)$,
- $\Rightarrow$Suite alternée de sommets et d'arcs :
  - $\gamma = x_0\ u_1\ x_1\ \ldots\ x_{k-1}\ u_k\ x_k$
- Tel que pour i de 1 à k,
  - x_{i-1} est extrémité initiale de l'arc u_i
  - x_i est son extrémité terminale.
- On dit que γ est un chemin de x_0 vers x_k de longueur k.

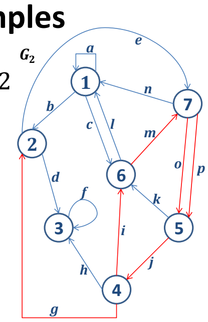
Chemin - Exemples

- $\gamma_1 = 6\ l\ 1\ a\ 1\ b\ 2\ d\ 3$
  - Chemin dans G_2
  - Allant de 6 vers 3
  - De longueur 4



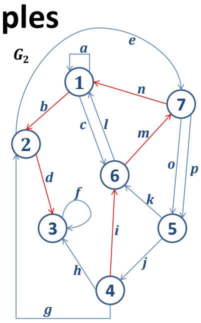
Chemin - Exemples

- $\gamma_2 = 7\ o\ 5\ j\ 4\ i\ 6\ m\ 7\ p\ 5\ j\ 4\ g\ 2$
  - Chemin dans G_2
  - Allant de 7 vers 2
  - De longueur 7



Chemin - Exemples

- $\gamma_3 = 4\ i\ 6\ m\ 7\ n\ 1\ b\ 2\ d\ 3$
  - Chemin dans G_2
  - Allant de 4 vers 3
  - De longueur 5



Propriétés Chaînes/Chemins

- Chaîne / Chemin simple
  - Si tous les arcs ou les arêtes les composant sont distincts.
- Chaîne / Chemin élémentaire
  - Si tous les sommets les composant sont distincts.

Propriétés Chaînes - Exemple

- $\mu_1 = 8\ l\ 4\ i\ 3\ e\ 2$
  - Simple et élémentaire
- $\mu_2 = 7\ q\ 6\ m\ 5\ j\ 4\ k\ 6\ m\ 5\ p\ 8\ l\ 4$
  - Non simple et non élémentaire
- $\mu_3 = 6\ b\ 1\ a\ 2\ d\ 3\ h\ 3\ i\ 4$
  - Simple mais non élémentaire

Propriétés Chaînes - Exemple

- $\mu_4 = 4\ g\ 2\ d\ 3\ h\ 4\ g\ 2\ b\ 1$
  - Non simple et non élémentaire
- $\mu_5 = 7\ o\ 5\ p\ 7\ n\ 1\ a\ 1\ c\ 6\ i\ 4$
  - Simple mais non élémentaire
- $\mu_6 = 6\ k\ 5\ j\ 4\ h\ 3\ d\ 2\ b\ 1\ n\ 7$
  - Simple et élémentaire

Propriétés Chemins - Exemple

- $\gamma_1 = 6\ l\ 1\ a\ 1\ b\ 2\ d\ 3$
  - Simple mais non élémentaire
- $\gamma_2 = 7\ o\ 5\ j\ 4\ i\ 6\ m\ 7\ p\ 5\ j\ 4\ g\ 2$
  - Non simple et non élémentaire
- $\gamma_3 = 4\ i\ 6\ m\ 7\ n\ 1\ b\ 2\ d\ 3$
  - Simple et élémentaire

Remarques (Chaînes/Chemins)

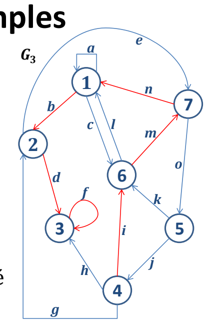
- Longueur d'1 chaîne (chemin) simple = nombre d'arêtes (arcs) formant cette chaîne (chemin).
- Si ∃ chemin d'1 sommet x vers 1 sommet y, on note : $x\ \alpha\ y$.
- Toute chaîne (ou chemin) élémentaire est aussi simple. L'inverse n'est pas toujours vrai.

Remarques (Chaînes/Chemins)

- Dans un graphe simple, une chaîne ou un chemin peuvent être déterminés juste en énumérant la suite des sommets qui les composent.
- Un chemin peut être déterminé juste en énumérant la suite des sommets qui le composent si le graphe est un 1-graphe

Chemin - Exemples

- $\gamma_4 = 4\ i\ 6\ m\ 7\ n\ 1\ b\ 2\ d\ 3\ f\ 3$
- $\gamma_4 = 4\ 6\ 7\ 1\ 2\ 3\ 3$
  - Chemin dans G_3
  - Allant de 4 vers 3
  - De longueur 5
  - Simple, non élémentaire
  - Il n'y a qu'une seule possibilité pour passer d'un sommet à un autre car c'est un 1-graphe

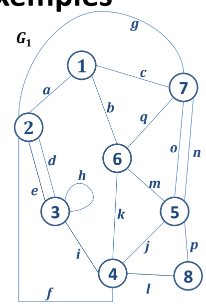


Chaîne fermée / Chemin fermé

- Un chemin dont les extrémités sont confondues est dit <u>chemin fermé</u>.
- Une chaîne dont les extrémités sont confondues est dite <u>chaîne fermée</u>.

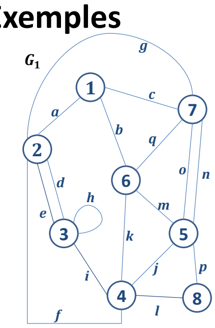
Chaîne fermée - Exemples

- $\mu_7 = 8\ l\ 4\ i\ 3\ e\ 2\ d\ 3\ i\ 4\ j\ 5\ p\ 8$
  - Chaîne fermée dans G_1
  - De longueur 7
  - Non simple, non élémentaire



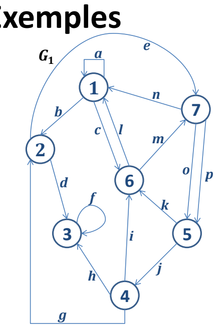
Chaîne fermée - Exemples

- $\mu_8 = 1\ b\ 6\ m\ 5\ o\ 7\ g\ 2\ a\ 1$
  - Chaîne fermée dans G_1
  - De longueur 5
  - Simple, non élémentaire



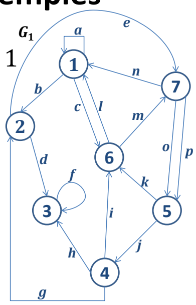
Chemin fermé - Exemples

- $\gamma_5 = 7\ n\ 1\ c\ 6\ m\ 7\ n\ 1\ b\ 2\ e\ 7$
  - Chemin fermée dans G_1
  - De longueur 6
  - Non simple, non élémentaire



Chemin fermé - Exemples

- $\gamma_6 = 1\ a\ 1\ b\ 2\ e\ 7\ o\ 5\ k\ 6\ m\ 7\ n\ 1$
  - Chemin fermée dans G_1
  - De longueur 7
  - Simple, non élémentaire

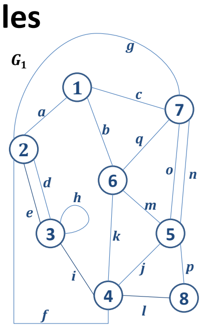


Cycle

- On appelle <u>cycle</u> dans un graphe non orienté (resp. orienté) $G=(X, E)$ (resp. $G=(X, U)$), toute chaîne fermée simple :
- $\mu = x_0\ e_1\ x_1\ \dots\ x_{k-1}\ e_k\ x_k$ (resp. $\mu = x_0\ u_1\ x_1\ \dots\ x_{k-1}\ u_k\ x_k$) Tel que $k > 0$, et $x_0 = x_k$.
- On dit que μ est un cycle de longueur k.

Cycle - Exemples

- $\mu_7 = 8\ l\ 4\ i\ 3\ e\ 2\ d\ 3\ i\ 4\ j\ 5\ p\ 8$
  - Chaîne fermée mais pas cycle
- $\mu_8 = 1\ b\ 6\ m\ 5\ o\ 7\ g\ 2\ a\ 1$
  - Cycle dans G_1
  - De longueur 5

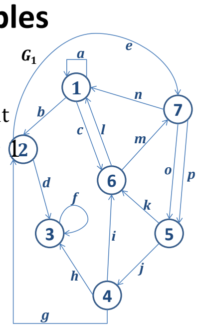


Circuit

- On appelle <u>circuit</u> dans un graphe orienté $G=(X, U)$, tout chemin fermé simple :
- $\gamma = x_0\ u_1\ x_1\ \dots\ x_{k-1}\ u_k\ x_k$ Tel que $k > 0$, et $x_0 = x_k$.
- On dit que μ est un circuit de longueur k.

Circuit - Exemples

- $\gamma_5 = 7\ n\ 1\ c\ 6\ m\ 7\ n\ 1\ b\ 2\ e\ 7$
  - Chemin fermée mais pas circuit
- $\gamma_6 = 1\ a\ 1\ b\ 2\ e\ 7\ o\ 5\ k\ 6\ m\ 7\ n\ 1$
  - Circuit dans G_1
  - De longueur 7



Cycle / Circuit élémentaire

- On dit qu'un cycle ou circuit est <u>élémentaire</u> si tous les sommets qui les composent sont distincts.
- On ne regarde pas la répétition due à la fermeture.
- Le cycle (resp. circuit) élémentaire est une chaîne (resp. un chemin) fermée non élémentaire.

Remarques (Cycles/Circuits) 1/2

- Longueur cycle ou circuit élémentaire = nombre de sommets formant ce cycle ou circuit.
- Dans un graphe simple, un cycle ou un circuit peuvent être déterminés juste en énumérant la suite des sommets qui les composent.

Remarques (Cycles/Circuits) 2/2

- Une boucle est un cycle élémentaire de longueur 1.
- Une boucle dans un graphe orienté est un circuit élémentaire de longueur 1.
- Tout cycle est aussi chaîne. Tout circuit est aussi chemin. Tout circuit est aussi cycle. Tout chemin est aussi chaîne.

Propositions

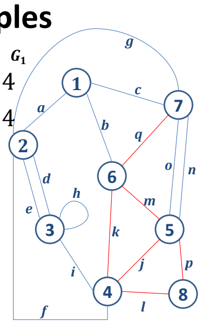
- Soit G=(X, E) un graphe non orienté. De toute chaîne joignant deux sommets x et $y \in X$, on peut extraire 1 chaîne élémentaire joignant x et y.
- Soit G=(X, U) un graphe orienté. De tout chemin allant du sommet $x \in X$ vers le sommet $y \in X$, on peut extraire 1 chemin élémentaire allant de x à y.
- Il suffit de supprimer les cycles (resp. circuits) intermédiaires

Propositions

- Soit G=(X, E) un graphe non orienté (resp. G=(X, U) un graphe orienté). De tout cycle (resp. circuit) passant par 1 arête $e \in E$ (resp. 1 arc $u \in U$), on peut extraire 1 cycle (resp. circuit) élémentaire passant par e (resp. u).
- Il suffit de supprimer les cycles (resp. circuits) intermédiaires qui ne passent pas par e (resp. par u)

Chaîne - Exemples

- μ_2 = 7 q 6 m 5 j 4 k 6 m 5 p 8 l 4
- μ_2 = 7 q 6 m 5 j 4 k 6 m 5 p 8 l 4
- μ_2' = 7 q 6 m 5 p 8 l 4



Existence d'un cycle

- Si G est un graphe vérifiant $\delta(G) \geq k \geq 2$ Alors G contient un cycle.
- Si de plus G est simple alors G admet un cycle élémentaire de longueur $\geq k$+1 et une chaine élémentaire de longueur $\geq k$.
- **Conséquence :**
  - Si $m \geq n$ (m étant le nombre d'arcs ou arêtes et n le nombre de sommets dans G) alors G admet un cycle.

Existence d'un circuit

- Si G est un graphe vérifiant $\delta^{+}(G) \geq k \geq 1$ (resp. $\delta^{-}(G) \geq k \geq 1$) Alors G contient un circuit.
- Si de plus G est simple alors G admet un circuit élémentaire de longueur $\geq k+1$ et un chemin élémentaire de longueur $\geq k$.

Matrice de fermeture transitive

- Soit $G=(X, U)$ un 1-graphe orienté d'ordre n.
- A partir de sa matrice d'adjacence M (doit être booléenne), on peut calculer la matrice de fermeture transitive de G
- Notée $\hat{M}$
- Chaque élément :

$$\hat{m}_{ij} = \begin{cases} 1 & si\ \exists i \alpha j \\ 0 & \text{sinon} \end{cases}$$

Calcul Matriciel Direct (1/2)

- On peut avoir la matrice de fermeture transitive par calcul matriciel comme suit :

$$\hat{M} = \bigvee_{l=1}^{n} M^{[l]}$$

- Où chaque matrice $M^{[l]}$ se calcule par récurrence (sur l) à travers le produit matriciel booléen comme suit :

$$\begin{cases} M^{[1]} = M \\ M^{[l+1]} = M^{[l]} * M \end{cases}$$

Calcul Matriciel Direct (2/2)

- Chaque élément de $M^{[l]}$:

$$m_{ij}^{[l]} = \bigvee_{k=1}^{n} (m_{ik}^{[l-1]} \wedge m_{kj})$$

  – où l varie de 2 à n.
- La matrice $M^{[l]}$ représente tous les chemins dans G de longueur l.

Algorithme de Warshall (1/2)

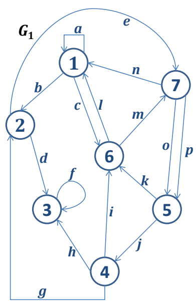
```
Algorithme Warshall
  Début
    Pour j de 1 à n Faire
        Pour i de 1 à n Faire
            Si M[i,j] = 1 Alors
              Pour k de 1 à n Faire
                M[i,k]  = M[i,k] ∨ M[j,k]
              Fait;
            fSi;
        Fait;
    Fait;
  Fin.
```


Algorithme de Warshall (2/2)

- Le calcul direct de $\hat{M}$ nécessite trop d'opérations matricielles.
- L'algorithme de Warshall permet un gain considérable en nombre d'opérations :
  n^2 tests et au plus n^3 opérations $\vee$,
- $\Rightarrow$ algorithme en $O(n^3)$

Exemple

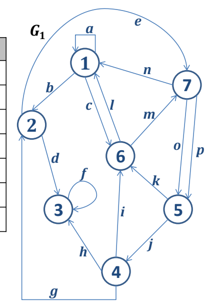
	1	2	3	4	5	6	7
1	1	1				1	
2			1				1
3			1				
4		1	1			1	
5				1		1	
6	1						1
7	1				2		



H. BENKAOUHA 43

Exemple

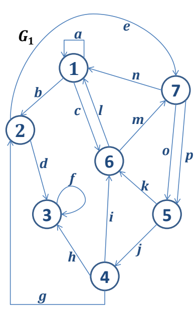
	1	2	3	4	5	6	7
1	1	1				1	
2			1				1
3			1				
4		1	1			1	
5				1		1	
6	1						1
7	1				1		



H. BENKAOUHA 44

Exemple

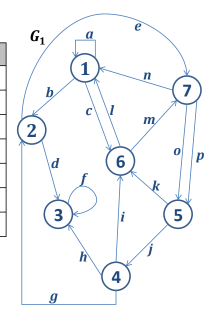
	1	2	3	4	5	6	7
1	1	1				1	
2			1				1
3			1				
4		1	1			1	
5				1		1	
6	1						1
7	1				1		



H. BENKAOUHA 45

Exemple

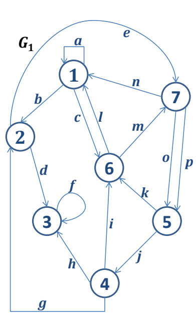
	1	2	3	4	5	6	7
1	1	1				1	
2			1				1
3			1				
4		1	1			1	
5				1		1	
6	1	1				1	1
7	1	1			1	1	



H. BENKAOUHA 46

Exemple

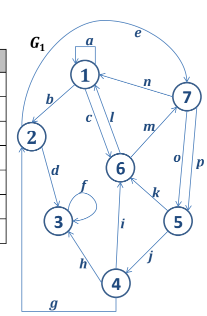
	1	2	3	4	5	6	7
1	1	1				1	
2			1				1
3			1				
4		1	1			1	
5				1		1	
6	1	1				1	1
7	1	1			1	1	



H. BENKAOUHA 47

Exemple

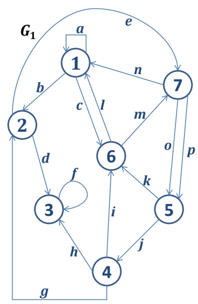
	1	2	3	4	5	6	7
1	1	1	1			1	1
2			1				1
3			1				
4		1	1			1	1
5				1		1	
6	1	1	1			1	1
7	1	1	1		1	1	1



H. BENKAOUHA 48

Exemple

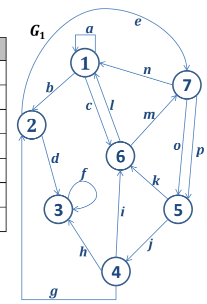
	1	2	3	4	5	6	7
1	1	1	1			1	1
2			1				1
3			1				
4		1	1			1	1
5				1		1	
6	1	1	1			1	1
7	1	1	1		1	1	1



H. BENKAOUHA 49

Exemple

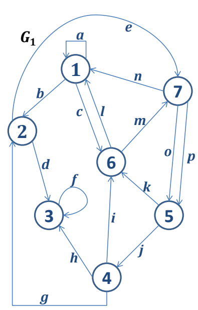
	1	2	3	4	5	6	7
1	1	1	1			1	1
2			1				1
3			1				
4		1	1			1	1
5				1		1	
6	1	1	1			1	1
7	1	1	1		1	1	1



H. BENKAOUHA 50

Exemple

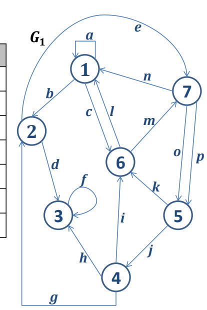
	1	2	3	4	5	6	7
1	1	1	1			1	1
2			1				1
3			1				
4		1	1			1	1
5		1	1	1		1	1
6	1	1	1			1	1
7	1	1	1		1	1	1



H. BENKAOUHA 51

Exemple

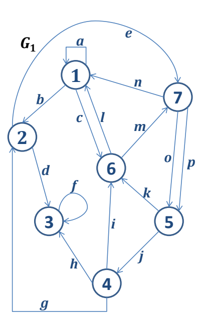
	1	2	3	4	5	6	7
1	1	1	1			1	1
2			1				1
3			1				
4		1	1			1	1
5		1	1	1		1	1
6	1	1	1			1	1
7	1	1	1		1	1	1



H. BENKAOUHA 52

Exemple

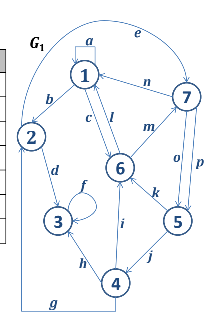
	1	2	3	4	5	6	7
1	1	1	1			1	1
2			1				1
3			1				
4		1	1			1	1
5		1	1	1		1	1
6	1	1	1			1	1
7	1	1	1	1	1	1	1



H. BENKAOUHA 53

Exemple

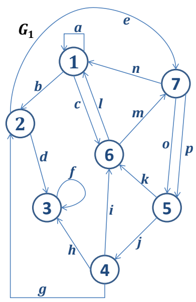
	1	2	3	4	5	6	7
1	1	1	1			1	1
2			1				1
3			1				
4		1	1			1	1
5		1	1	1		1	1
6	1	1	1			1	1
7	1	1	1	1	1	1	1



H. BENKAOUHA 54

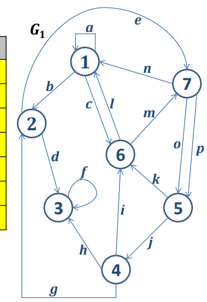
Exemple

	1	2	3	4	5	6	7
1	1	1	1			1	1
2			1				1
3			1				
4	1	1	1			1	1
5	1	1	1	1		1	1
6	1	1	1			1	1
7	1	1	1	1	1	1	1



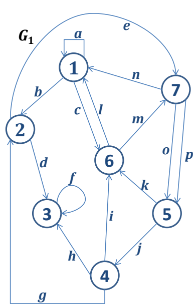
Exemple

	1	2	3	4	5	6	7
1	1	1	1			1	1
2			1				1
3			1				
4	1	1	1			1	1
5	1	1	1	1		1	1
6	1	1	1			1	1
7	1	1	1	1	1	1	1



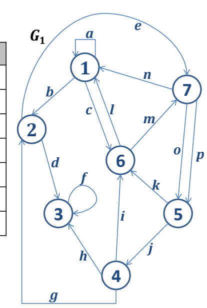
Exemple

	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1
3			1				
4	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1



Exemple

	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1
3	0	0	1	0	0	0	0
4	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1



Exploration (Parcours) d'un graphe

- L'exploration d'un graphe est un parcours (via les arcs ou les arêtes)
- Permettant d'examiner de façon exhaustive (visiter) les sommets.
- L'exploration d'un graphe permet d'étudier une ou plusieurs propriétés du graphe tel que :
  - la connexité, la forte connexité, biparti, ...

Algorithme d'exploration (1/2)

- Principe :
  - Consiste à déterminer l'ordre dans lequel seront visités les sommets.
  - Le parcours commence d'un sommet de départ *r* qu'on appelle *racine*
  - Il donne comme résultat une liste ordonnée de sommets où *r* apparaît en premier et les autres sommets apparaissent une seule fois.

Algorithme d'exploration (2/2)

```
P ← ∅ ;
L ← {r} ;
Tant que ((L ≠ ∅) et (P ≠ X))  Faire
  Choisir_extraire (i∈L) ;
  Pour (tout (i, j)∈U)   Faire
    Si (j∉P) Alors Ajouter j à L ; fSi;
  Fait;
  Pour (tout (j, i)∈U)   Faire
    Si (j∉P) Alors Ajouter j à L ; fSi;
  Fait;
  Ajouter i à la fin de P ;
Fait
```


Remarques : Algo. d'exploration

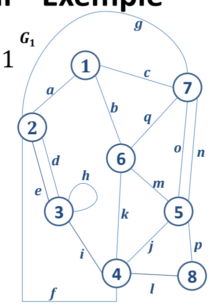
- Nous supposons que la fonction **`choisir_extraire`** existe et qui consiste à choisir de façon déterministe un sommet de *L* puis le supprime de *L*.
- Nous expliquons après les différentes implémentations de cette fonction.
- Il est possible d'appliquer cet algorithme sur un graphe non-orienté.

Exploration en largeur

- Consiste à parcourir le graphe
  - à partir du sommet de départ la racine (*r*)
  - puis ses voisins
  - puis les voisins des voisins non explorés
  - et ainsi de suite jusqu'à la fin.
- Nous pouvons utiliser l'algorithme d'exploration en déclarant
  - *L* comme une liste FIFO (premier arrivé, premier sorti)
  - La fonction **`choisir_extraire`** devient **`defiler`**
  - La fonction **`Ajouter`** devient **`enfiler`**.

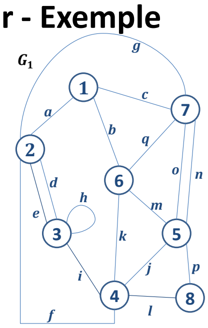
Exploration en largeur - Exemple

- Sommet de départ (racine) : 1
- File : 1
- Parcours :



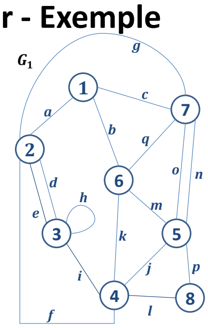
Exploration en largeur - Exemple

- File : 2 6 7
- Parcours : 1

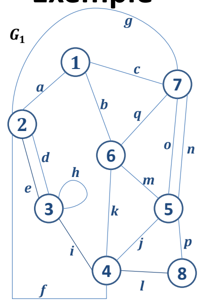


Exploration en largeur - Exemple

- File : 6 7 3 4 7
- Parcours : 1 2

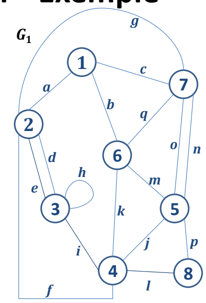


Exploration en largeur - Exemple



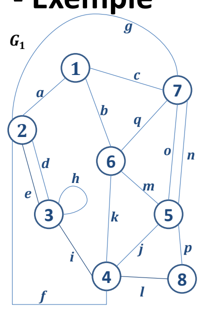
- File : 7 3 4 7 4 5 7
- Parcours : 1 2 6

Exploration en largeur - Exemple



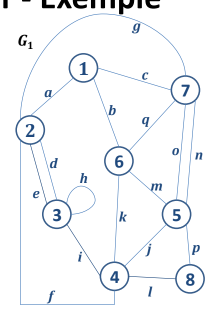
- File : 3 4 7 4 5 7 5
- Parcours : 1 2 6 7

Exploration en largeur - Exemple



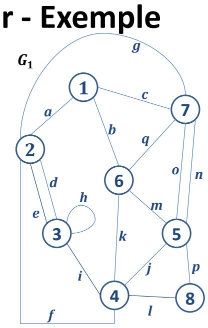
- File : 4 7 4 5 7 5 3 4
- Parcours : 1 2 6 7 3

Exploration en largeur - Exemple



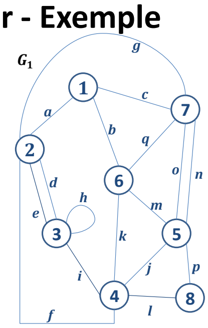
- File : 7 4 5 7 5 3 4 5 8
- Parcours : 1 2 6 7 3 4

Exploration en largeur - Exemple



- File : 4 5 7 5 3 4 5 8
- Parcours : 1 2 6 7 3 4

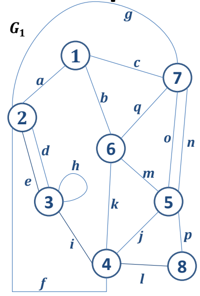
Exploration en largeur - Exemple



- File : 5 7 5 3 4 5 8
- Parcours : 1 2 6 7 3 4

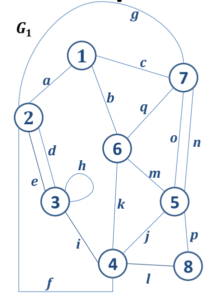
Exploration en largeur - Exemple

- File : 7 5 3 4 5 8 8
- Parcours : 1 2 6 7 3 4 5



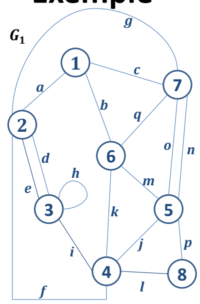
Exploration en largeur - Exemple

- File : 5 3 4 5 8 8
- Parcours : 1 2 6 7 3 4 5



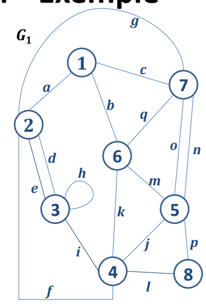
Exploration en largeur - Exemple

- File : 3 4 5 8 8
- Parcours : 1 2 6 7 3 4 5



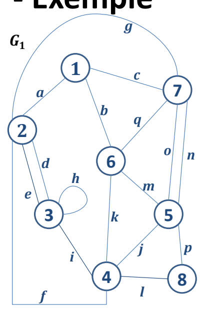
Exploration en largeur - Exemple

- File : 4 5 8 8
- Parcours : 1 2 6 7 3 4 5



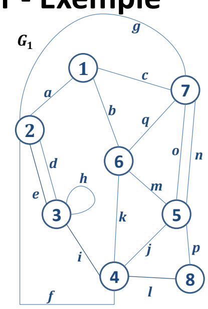
Exploration en largeur - Exemple

- File : 5 8 8
- Parcours : 1 2 6 7 3 4 5



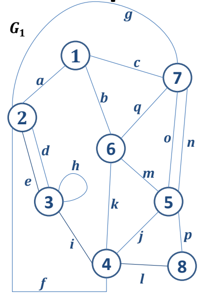
Exploration en largeur - Exemple

- File : 8 8
- Parcours : 1 2 6 7 3 4 5



Exploration en largeur - Exemple

- File : 8
- Parcours : 1 2 6 7 3 4 5 8
- Fin du parcours,
  tous les sommets sont dans
  le parcours.

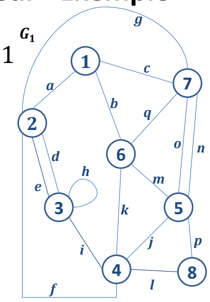


Exploration en profondeur

- Consiste à parcourir le graphe
  - à partir du sommet de départ la racine (r)
  - puis tracer une chaîne à partir de ce sommet
  - puis choisir un autre sommet (parmi les voisins des sommets de cette chaîne dans l'ordre) et faire de même jusqu'à la fin.
- Nous pouvons utiliser l'algorithme d'exploration en déclarant
  - L comme une pile (dernier arrivé, premier sorti)
  - La fonction `choisir_extraire` devient `depiler`
  - La fonction `Ajouter` devient `empiler`.

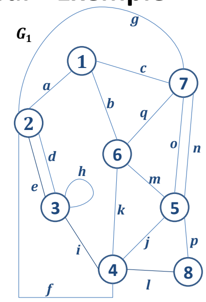
Exploration en profondeur - Exemple

- Sommet de départ (racine) : 1
- Pile : 1
- Parcours :



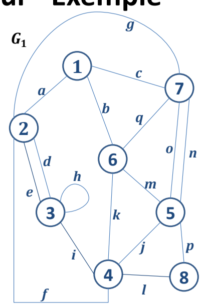
Exploration en profondeur - Exemple

- Pile : 2 6 7
- Parcours : 1



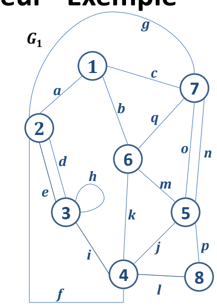
Exploration en profondeur - Exemple

- Pile : 2 6 2 5 6
- Parcours : 1 7

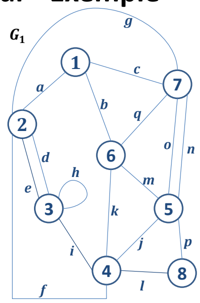


Exploration en profondeur - Exemple

- Pile : 2 6 2 5 4 5
- Parcours : 1 7 6

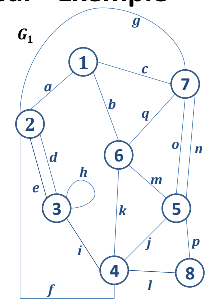


Exploration en profondeur - Exemple



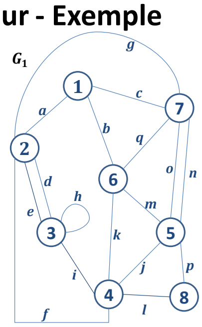
- Pile : 2 6 2 5 4 4 8
- Parcours : 1 7 6 5

Exploration en profondeur - Exemple



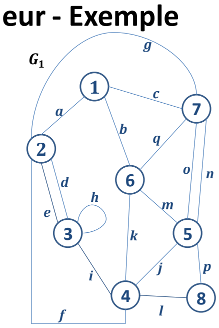
- Pile : 2 6 2 5 4 4 4
- Parcours : 1 7 6 5 8

Exploration en profondeur - Exemple



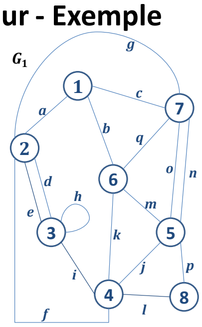
- Pile : 2 6 2 5 4 4 2 3
- Parcours : 1 7 6 5 8 4

Exploration en profondeur - Exemple



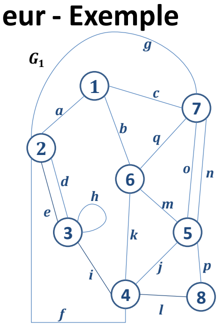
- Pile : 2 6 2 5 4 4 2 2 3
- Parcours : 1 7 6 5 8 4 3

Exploration en profondeur - Exemple



- Pile : 2 6 2 5 4 4 2 2
- Parcours : 1 7 6 5 8 4 3

Exploration en profondeur - Exemple



- Pile : 2 6 2 5 4 4 2
- Parcours : 1 7 6 5 8 4 3 2
- Fin du parcours,
  tous les sommets sont dans
  le parcours.

Connexité

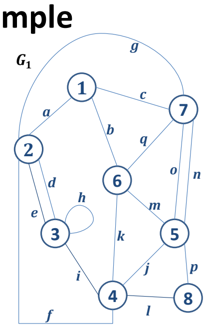
- Un graphe est dit connexe s'il existe
  - une chaîne joignant chaque paire de sommets x et y $(x \neq y)$.
- Dessin :
  - On le voit comme une seule entité.

Composante Connexe (CC)

- Soit un graphe $G = (X, E)$ (resp. $G = (X, U)$) :
  - Le sous graphe engendré par un sommet isolé est considéré comme une composante connexe de G.
  - Si le sous graphe engendré par un ensemble de sommets $S \subseteq X$ (G_S) est connexe et tout sous graphe engendré par $S \cup \{x\}$ et $x \notin S$ n'est pas connexe Alors G_S est une composante connexe de G.
- Un graphe connexe contient une seule composante connexe.

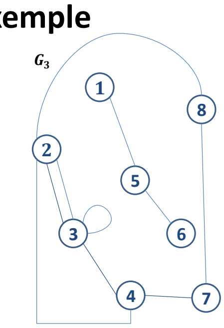
Connexité - Exemple

- Graphe connexe
- 1 CC = X



Connexité - Exemple

- Graphe non connexe
- 2 CC
  - $C_1 = \{1, 5, 6\}$
  - $C_2 = \{2, 3, 4, 7, 8\}$



Algorithme de connexité (1/2)

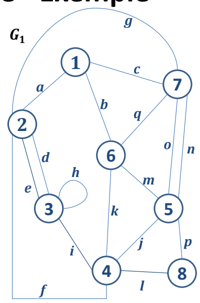
- Nous pouvons utiliser l'algorithme d'exploration afin de vérifier la connexité.
- Il s'agit juste de vérifier si la sortie $P = X$.
- Il existe aussi d'autres algorithmes permettant de vérifier la connexité.
- Le suivant utiliser les marquages.

Algorithme de connexité (2/2)

```
C ← {r} ;
Pour (tout i ∈ X)  Marque[i] ← faux ;
Tant que (∃ i∈C tel que Non(Marque[i]))
  Pour (tout (i, j)∈U)
    C ← C ∪ {j} ;
  Pour (tout (j, i)∈U)
    C ← C ∪ {j} ;
  Marque[i] ← vrai ;
Si C=X Alors Connexe ← Vrai ;
       Sinon Connexe ← Faux ;
```

Algorithme de Connexité - Exemple

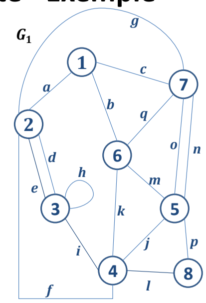
- Graphe connexe
- $C = \{\}$



H. BENKAOUHA 97

Algorithme de Connexité - Exemple

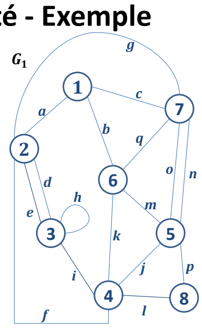
- Ajouter à C un sommet quelconque
- On choisit le 1
- $C = \{1\}$



H. BENKAOUHA 98

Algorithme de Connexité - Exemple

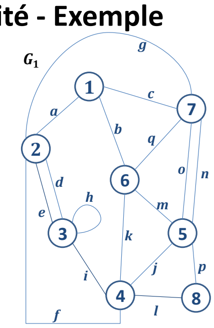
- Ajouter Voisins de 1
- $C = \{1, 2, 6, 7\}$



H. BENKAOUHA 99

Algorithme de Connexité - Exemple

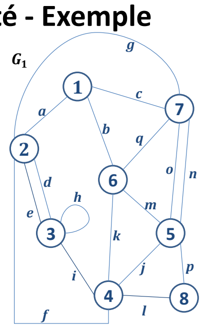
- Ajouter Voisins de 2
- $C = \{1, 2, 6, 7, 3, 4\}$



H. BENKAOUHA 100

Algorithme de Connexité - Exemple

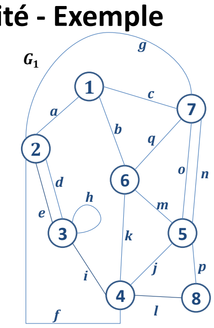
- Ajouter Voisins de 6
- $C = \{1, 2, 6, 7, 3, 4, 5\}$



H. BENKAOUHA 101

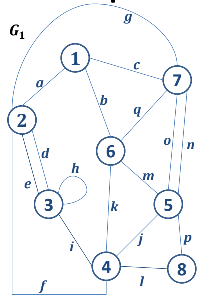
Algorithme de Connexité - Exemple

- Ajouter voisins de 4
- $C = \{1, 2, 6, 7, 3, 4, 5, 8\}$



H. BENKAOUHA 102

Algorithme de Connexité - Exemple



- Ajouter voisins de 4
- $C = X$
- Fin de l'Algorithme
- G_1 est connexe

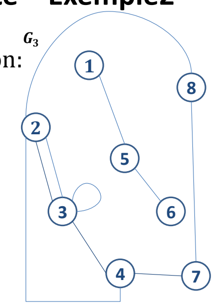
H. BENKAOUHA 103

Algorithme de calcul des CC

```
  k ← 1;
  Tant que X ≠ ϕ
    Faire
      Choisir r ∈ X;
      C[k] ← Connexité (G, r);
// Ici, on considère que l'algorithme précédent ou
// l'algorithme d'exploration comme fonction qui a
// en entrée le graphe et un sommet de départ et
// retourne une CC C ou le parcoures P
      X ← X – C[k];
      k ← k + 1;
    Fait;
```

H. BENKAOUHA 104

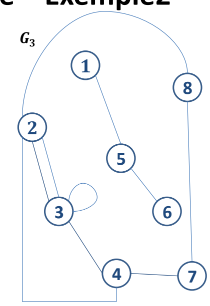
Algorithme de Connexité – Exemple2



- L'exploration comme fonction:
- File : 1
- Parcours :

H. BENKAOUHA 105

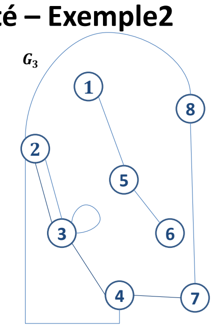
Algorithme de Connexité – Exemple2



- File : 5
- Parcours : 1

H. BENKAOUHA 106

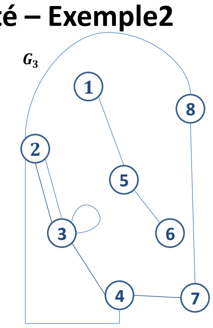
Algorithme de Connexité – Exemple2



- File : 6
- Parcours : 1 5

H. BENKAOUHA 107

Algorithme de Connexité – Exemple2

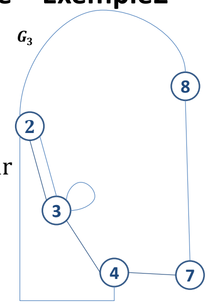


- File :
- Parcours : 1 5 6
- Fin, car File vide
- G n'est pas connexe

H. BENKAOUHA 108

Algorithme de Connexité – Exemple2

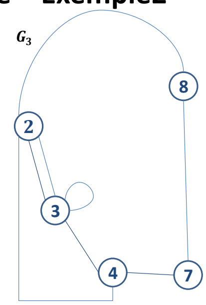
- C_1 = { 1, 5, 6 }
- X = {2, 3, 4, 7, 8}
- On refait l'exploration à partir d'un sommet de X.



H. BENKAOUHA 109

Algorithme de Connexité – Exemple2

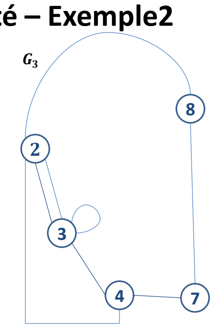
- File : 2
- Parcours :



H. BENKAOUHA 110

Algorithme de Connexité – Exemple2

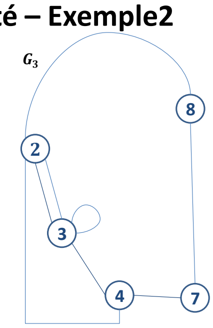
- File : 3 4 8
- Parcours : 2



H. BENKAOUHA 111

Algorithme de Connexité – Exemple2

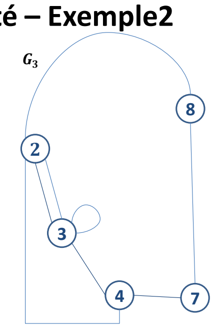
- File : 4 8 3 4
- Parcours : 2 3



H. BENKAOUHA 112

Algorithme de Connexité – Exemple2

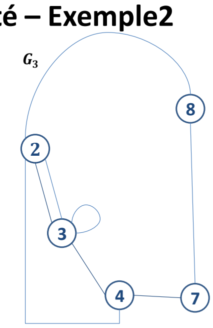
- File : 8 3 4 7
- Parcours : 2 3 4



H. BENKAOUHA 113

Algorithme de Connexité – Exemple2

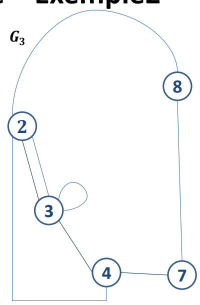
- File : 3 4 7 7
- Parcours : 2 3 4 8



H. BENKAOUHA 114

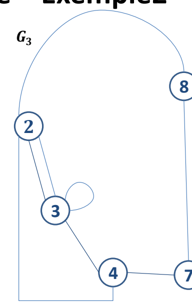
Algorithme de Connexité – Exemple2

- File : 4 7 7
- Parcours : 2 3 4 8



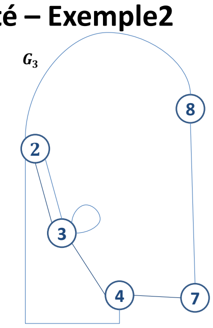
Algorithme de Connexité – Exemple2

- File : 7 7
- Parcours : 2 3 4 8



Algorithme de Connexité – Exemple2

- File : 7
- Parcours : 2 3 4 8 7
- Fin, car P contient tous les sommets restants dans X



Algorithme de Connexité – Exemple2

G_3

- $C_2 = \{2, 3, 4, 7, 8\}$
- $X = \{\}$
- On s'arrête car X est vide.
- On a 2 CC.

Forte Connexité

- Un graphe orienté $G=(X, U)$ est <u>fortement connexe</u> (f.c.) s'il existe entre chaque paire de sommets x et $y \in X$ $(x \neq y)$:
  - **un chemin de x à y ($x \alpha y$)**

et

  - **un chemin de y à x ($y \alpha x$)**.

Composante Fortement Connexe (CFC)

- Soit $G=(X, U)$ un graphe orienté :
  - Le sous graphe engendré par un sommet $x \in X$ tel que $d_G^+(x) = 0$ ou $d_G^-(x) = 0$ forme une composante fortement connexe de G.
  - Si le sous graphe engendré par un ensemble de sommets $S \subseteq X$ (G_S) est fortement connexe et le sous graphe engendré par $S \cup \{x\}$ et $x \notin S$ n'est pas fortement connexe Alors G_S est une composante fortement connexe de G.

Ascendants / Descendants

- Soit $G=(X, U)$ un graphe orienté,
- On définit pour chaque sommet $x \in X$,
- 2 ensembles :
  - L'ensemble des descendants de x :
    $D(x) = \{ y \in X \,/\, x \,\alpha\, y \}$
  - L'ensemble des ascendants de x :
    $A(x) = \{ y \in X \,/\, y \,\alpha\, x \}$

Algorithme de calcul des CFCs

```
D ← {r} ;
Pour (tout i ∈ X)   Marque[i] ← faux ;
Tant que ((∃ i ∈ D) et (Marque[i]=faux) )
   Marque[i] ← vrai ;
   Pour (tout (i, j) ∈ U )   D ← D ∪ {j} ;
A ← {r} ;
Pour (tout i ∈ X)   Marque[i] ← faux ;
Tant que ((∃ i ∈ A) et (Marque[i]=faux))
   Marque[i] ← vrai ;
   Pour (tout (j, i) ∈ U)   A ← A ∪ {j}
CFC  ← D ∩ A
```


Algorithme de calcul des CFCs à partir de la matrice de fermeture transitive

```
Tout sommet i ayant mii=0
   seul dans une CFC
Les autres sommets
   Sommets ayant :
   lignes identiques
        et
   colonnes identiques
   dans la même CFC
```


Exemple - CFC

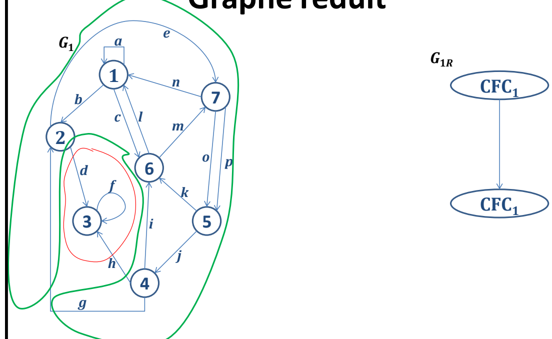
	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1
3	0	0	1	0	0	0	0
4	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1

- Lignes identiques :
  $L_1 = \{1, 2, 4, 5, 6, 7, 8\}$
  $L_2 = \{3\}$
- Colonnes identiques :
  $C_1 = \{1, 2, 4, 5, 6, 7, 8\}$
  $C_2 = \{3\}$
- Les CFCs:
  $CFC_1 = \{1, 2, 4, 5, 6, 7, 8\}$
  $CFC_2 = \{3\}$

Graphe réduit

- A tout graphe orienté $G=(X, U)$ on associe le graphe simple $G_R=(X_R, U_R)$ appelé <u>graphe réduit</u> de G défini comme suit :
  - $X_R = \{$ A chaque c.f.c. de G correspond un sommet $C_i \}$
  - $U_R = \{ (C_i, C_j) \,/\, i \neq j$ et $\exists\, x \in C_i$ et $\exists\, y \in C_j$ et $(x, y) \in U \}$
- Un graphe fortement connexe possède une seule C.F.C.
- Le graphe réduit d'un graphe ne possède pas de circuits.

Graphe réduit



Parcours Euleriens

- Un parcours Eulerien passe une fois et une seule fois par chaque arête (resp. arc) du graphe.
- Le parcours peut être une chaîne, un chemin, un cycle ou un circuit.
- Soit G un graphe contenant m arêtes (resp. m arcs) :
- Une chaîne simple, un chemin simple, un cycle ou un circuit de longueur m est appelé Eulérien.

Théorème d'Euler (1766)

- Un multigraphe G admet une <u>chaîne Eulérienne</u>
- Si et seulement si
  - il est connexe (à des sommets isolés près)

  et
  - le nombre de sommets de degré impair est 0 ou 2.

Théorème d'Euler (1766)

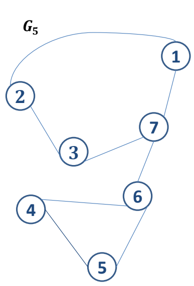
- **Conséquences**
  - Un graphe G admet une chaîne Eulérienne d'un sommet x à un sommet y ($x \neq y$) si et seulement si $d_G(x)$ et $d_G(y)$ sont impairs et $\forall z$ sommet de G ($z \neq x$ et $z \neq y$), on a $d_G(z)$ pair.
  - Un graphe G admet un cycle Eulérien si et seulement $\forall x$ sommet de G, on a $d_G(x)$ pair.

Détermination d'une chaîne Eulerienne

- Choisir sommet a de degré impair (Si pas de sommets de degrés impairs, choisir n'importe quel sommet).
  - On construit une chaîne à partir de a comme suit :
- A chaque étape k
  - On obtient une chaîne de longueur k
  - G_k correspond au graphe partiel de G engendré par l'ensemble des arêtes (resp. d'arcs) initial auquel on supprime ceux faisant partie de la chaîne.
- A chaque étape k, en arrivant à un sommet x,
  - Il faut éviter de prendre toute arête (resp. arc) qui est isthme dans G_k.
  - Sauf s'il s'agit de la seule et unique possibilité, on la prend.
- G_k graphe constitué de sommets isolés $\Rightarrow$ Fin.

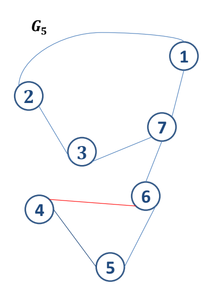
Exemple

- Sommets de degrés impairs :
  - 6 et 7
- $\mu = 6$



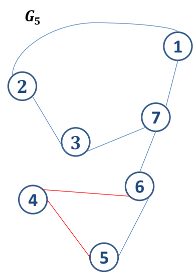
Exemple

- $\mu = 6$
- On a {6,7} ou {6,4} ou {6,5}
- {6,7} déconnecte le graphe, on la prend pas
- On prend par exemple {6,4}
- $\mu = 6$ 4



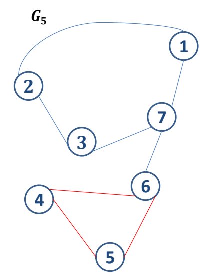
Exemple

- $\mu = 6\ 4$
- On a {4,5}
- On la prend
- $\mu = 6\ 4\ 5$



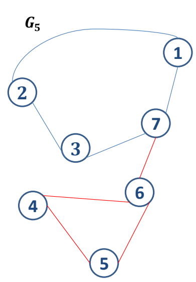
Exemple

- $\mu = 6\ 4\ 5$
- On a {5,6}
- On la prend
- $\mu = 6\ 4\ 5\ 6$



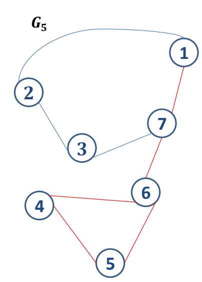
Exemple

- $\mu = 6\ 4\ 5\ 6$
- On a {6,7}
- On la prend
- $\mu = 6\ 4\ 5\ 6\ 7$



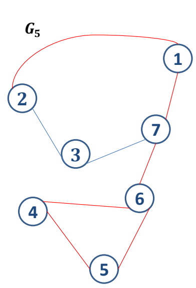
Exemple

- $\mu = 6\ 4\ 5\ 6\ 7$
- On a {7,1} ou {7,3}
- On prend par exemple {7,1}
- $\mu = 6\ 4\ 5\ 6\ 7\ 1$



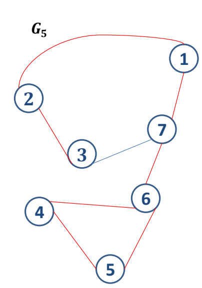
Exemple

- $\mu = 6\ 4\ 5\ 6\ 7\ 1$
- On a {1,2}
- On la prend
- $\mu = 6\ 4\ 5\ 6\ 7\ 1\ 2$



Exemple

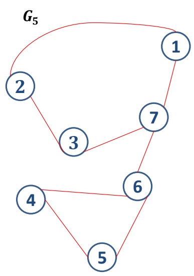
- $\mu = 6\ 4\ 5\ 6\ 7\ 1\ 2$
- On a {2,3}
- On la prend
- $\mu = 6\ 4\ 5\ 6\ 7\ 1\ 2\ 3$



Exemple

- $\mu = 6\ 4\ 5\ 6\ 7\ 1\ 2\ 3$
- On a {3,7}
- On la prend
- $\mu = 6\ 4\ 5\ 6\ 7\ 1\ 2\ 3\ 7$

Chaîne Eulérienne



Circuit Eulérien

- **Proposition**
  - Un graphe $G=(X, U)$ admet un circuit Eulérien

  Si et seulement si
  - Pour tout sommet x, on a $d^+_G(x)= d^-_G(x)$.
  - On dit que G est pseudo-symétrique.

Graphe Eulérien / semi-Eulérien

- G admet un cycle Eulérien

$\Rightarrow$ G est Eulérien.
- G admet une chaîne Eulérienne mais pas de cycle Eulérien

$\Rightarrow$ G est semi-Eulérien.

Parcours Hamiltonien (1/2)

- Un parcours Hamiltonien passe une fois et une seule fois par chaque sommet du graphe.
- Le parcours peut être une chaîne, un chemin, un cycle ou un circuit.

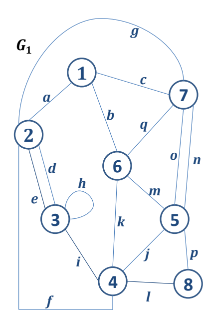
Parcours Hamiltonien (2/2)

- Soit G un graphe d'ordre n :
- Une chaîne (resp. un chemin) élémentaire de longueur n-1 est appelé chaîne Hamiltonienne (resp. chemin Hamiltonien).
- Un cycle (resp. circuit) élémentaire de longueur n est appelé cycle (resp. circuit) Hamiltonien.

Exemple

- 1 a 2 e 3 i 4 k 6 q 7 o 5 p 8
- Chaîne Hamiltonienne

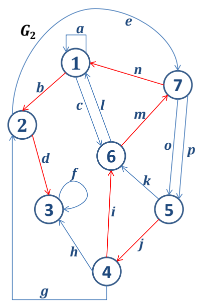
- 1 a 2 d 3 i 4 l 8 p 5 o 6 p 7 c 1
- Cycle Hamiltonien



Exemple

- $5\ j\ 4\ i\ 6\ m\ 7\ n\ 1\ b\ 2\ d\ 3$
- Chemin Hamiltonien

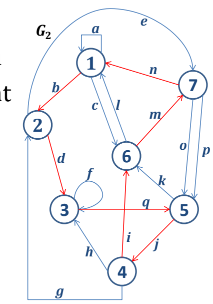
- Il n'y a pas de circuit Hamiltonien car G n'est pas fortement connexe.



Exemple

- Si on rajoute un arc (3, 5) au graphe précédent, on obtient Un circuit Hamiltonien

- $5\ j\ 4\ i\ 6\ m\ 7\ n\ 1\ b\ 2\ d\ 3\ q\ 5$



Graphe Hamiltonien / semi-Hamiltonien

- Un graphe qui contient un cycle Hamiltonien $\Rightarrow$ graphe Hamiltonien.
- Un graphe semi-Hamiltonien : contient une chaîne Hamiltonienne, mais pas de cycle Hamiltonien.
- Le plus petit graphe Hamiltonien d'ordre n est le graphe cycle (Graphe connexe non-orienté à n arêtes. Il est 2-régulier)

Graphe Hamiltonien - Propositions

- Un graphe complet d'ordre $n \geq 3$ est Hamiltonien.
- Tout graphe tournoi (un graphe orienté simple et complet) d'ordre n, noté T_n contient un chemin Hamiltonien.
- Tout tournoi d'ordre n (T_n) fortement connexe contient un circuit Hamiltonien.

Graphe Hamiltonien - Théorème

- Utilisé pour démontrer qu'un graphe n'est pas Hamiltonien (ne contient pas de cycle Hamiltonien).
- Si $G=(X,E)$ est un graphe Hamiltonien, alors pour tout ensemble de sommets $S \subset X$, on a :
  - $p(G_{X-S}) \leq |S|$
  - où $p(G_{X-S})$ est le nombre de composantes connexes du sous graphe de G induit par l'ensemble X-S