Introduction à l’informatique – Graphes II

Table des matières

[RAPPELS 3](#_Toc530221963)

[DEFINITIONS CLASSIQUES 3](#_Toc530221964)

[CONNEXITE (FORTE) 6](#_Toc530221965)

[CHAINES ET CYCLES EULERIENS 9](#_Toc530221966)

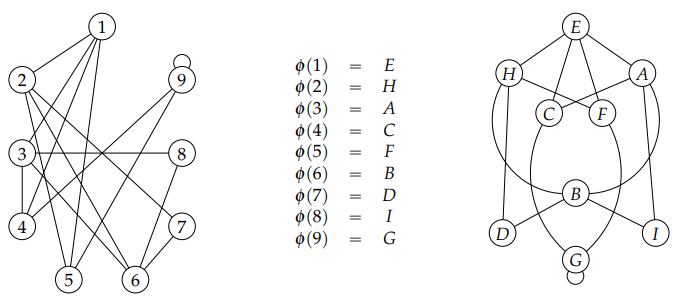
[COLORATION 10](#_Toc530221967)

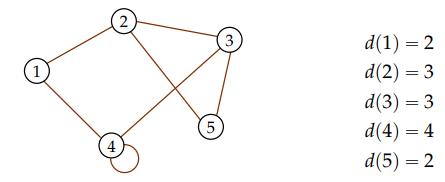
# RAPPELS

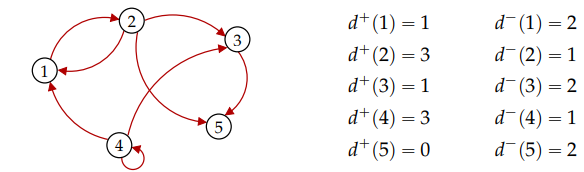
* Soit G = (S, A) un graphe non orienté et a = {u, v} ∈ A une de ses arêtes. A est incidente à u et v. u et v sont voisins et adjacents.
* Soit G = (S, A) un graphe orienté et a = (u, v) ∈ A une de ses arcs. A est incident à u et à v. u et v sont adjecents. V(resp. u) est voisin sortant (resp. entrant) de u (resp. v) et tel que v ∈ |+ (u) (resp. u ∈ |- (v)) -> diapo page 4
* Soit G = (S, A) un graphe. La matrice adjacente de G, notée B(G), est la matrice booléenne d’ordre |V| tq B(G)\_(i,j) = {1 si {i, j} ou (i, j) ∈ A | 0 sinon}

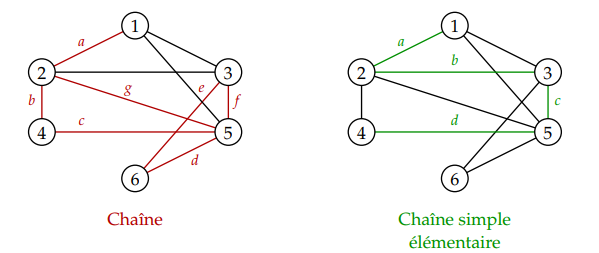
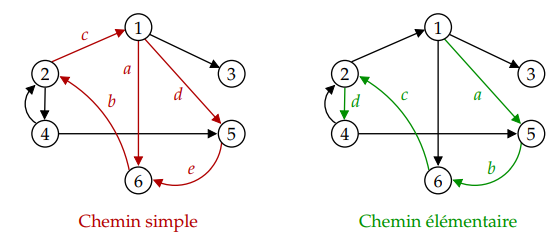
# DEFINITIONS CLASSIQUES

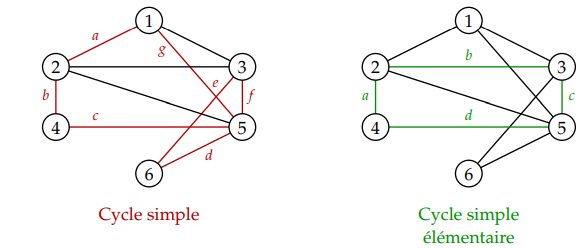
* Deux graphes G = (S, A) et G’ = (S’, A’) sont isomorphes s’il existe une bijection φ : S -> S’ telle que (u, v) ∈ A ⬄ (φ (u), φ (v)) ∈ A’

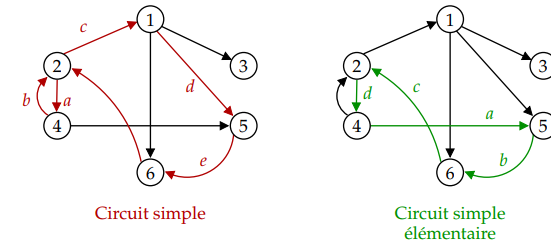


* Soit G = (S, A) un graphe non orienté et s ∈ S. Le degré (ou valence) de s, noté d(s) est le nombre d’aretes incidentes à s, les boucles étant comptées deux fois, et telle que Somme\_(s ∈ S) deg(s) = 2\*|a|  
  
* Soit G = (S, A) un graphe orienté et s ∈ S. Le degré sortant (resp. entrant) de u, noté d+(u) (resp. d-(u)) est le nombre d’arcs sortant/entrant dans ce sommet, cad d+(u) = |Γ+(u)|(resp. d-(u) = |Γ+(u)|)



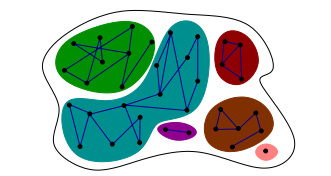
* Soit G = (S, A) un graphe non orienté. Une chaîne de u à v, notée µ (u, v), est une suites d’aretes consécutives reliant u à v.
* Une chaine est simple/élémentaire) si elle ne passe pas deux fois par la meme arete/meme sommet.  
  
* Soit G = (S, A) un graphe orienté. Un chemin de u à v, noté µ[u, v], est une suite d’arcs consécutifs reliant u à v
* Un chemin est simple/élémentaire s’il ne passe pas deux fois par le même arc/meme sommet.  
  
* Soit G = (S, A) un graphe non orienté. Un cycle de G est une chaine dont les extrémités coïncident
* Un cycle est simple (resp. élémentaire) s’il est une chaîne simple (resp. élémentaire).



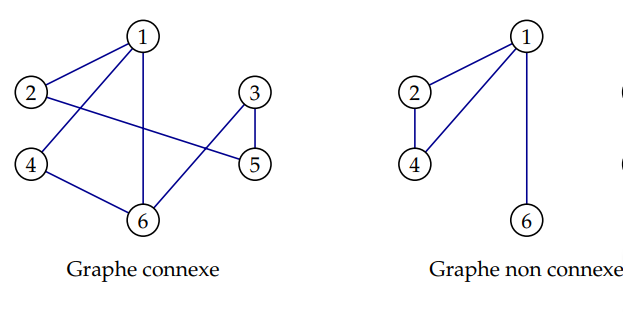
* Soit G = (S, A) un graphe orienté. Un cycle de G est un circuit dont les extremités coïncident
* Un circuit est simple/élémentaire s’il est un chemin simple/élementaire  
  

# CONNEXITE (FORTE)

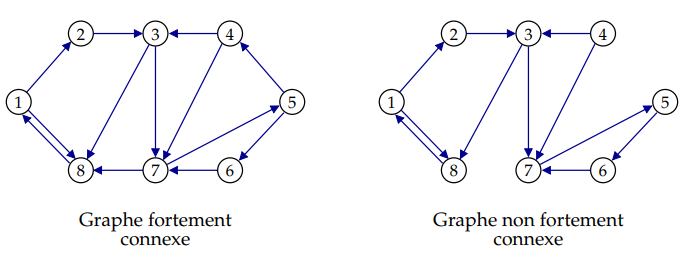
Equivalence :

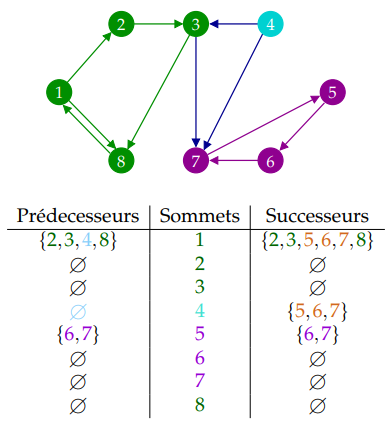
* Soit E un ensemble et R une relation de E dans E, R est une relation d’équivalence si et seulement si elle est réflexive, symétrique et transitive.
* Soit E un ensemble, x un élément de E, et R une relation d’équivalence de E dans E. La classe d’équivalence [x] est l’ensemble des y ∈  E tels que :  
  y  ∈ [x] ⬄ xRy  
  

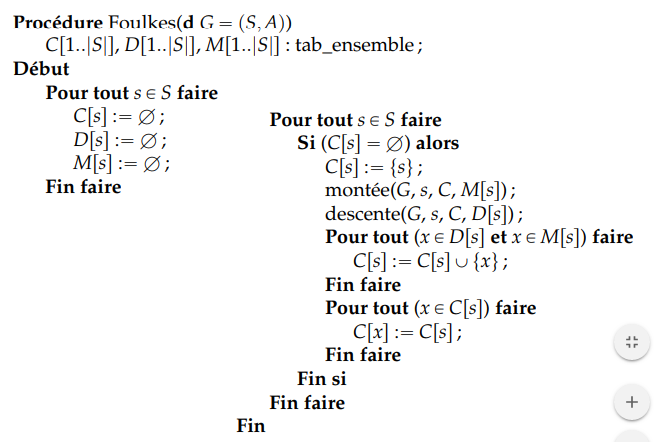
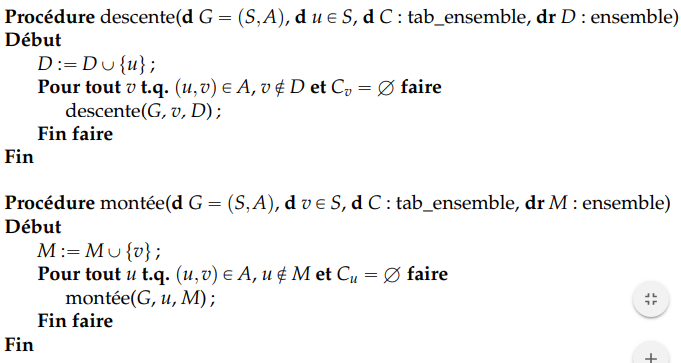
Connexité

* Un graphe G = (S, A) est connexe si, ∀u, v ∈ S, ∃µ(u, v)
* La relation uRv ⬄ (u = v) ou (∃µ(u, v)), est une relation d’equivalence et les classes d’equivalence induites sur S forment une partition {S\_1,…,S\_p}
* Les sous-graphes C\_1,…,C\_p engendrés par les sous-ensembles S\_1,…,S\_p sont les composantes connexes de G.  
  

Connexité (forte)

* Un graphe orienté G = (S, A) est fortement connexe si, ∀u, v ∈ S, ∃µ[u, v] et ∃µ[v, u]
* La relation uRv ⬄ (u = v) ou (∃µ[u, v] et ∃µ[v, u]) est une relation d’équivalence et les classes d’équivalence induites sur S forment une partition {S\_1,...,S\_P}.
* Les sous-graphes C\_1,…,C\_p engendrés par les sous-ensembles S\_1,…,S\_p sont les composantes connexes de G.  
  

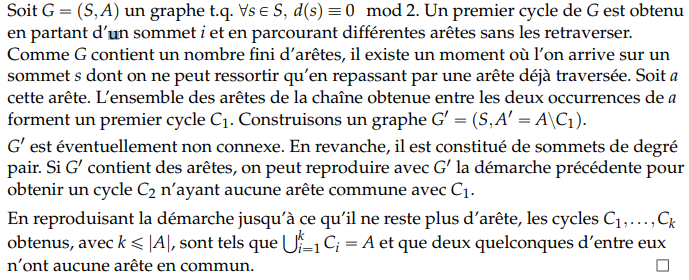
Calcul des composantes fortement connexes :  


* Fortement connexe si ils sont predesseseur et succeseurs d’un sommet -> on n’aurait pas besoin de calculer le sommet de ces nombres car déjà dans une fortement connexe
* Quand predesseseur et successeur d’un sommet sont fortement connexe et déjà calculer dans le tableau on ne les ets pas car déjà calculer.
* Algorithme de Foulkes :  
    
  

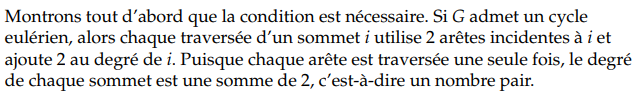
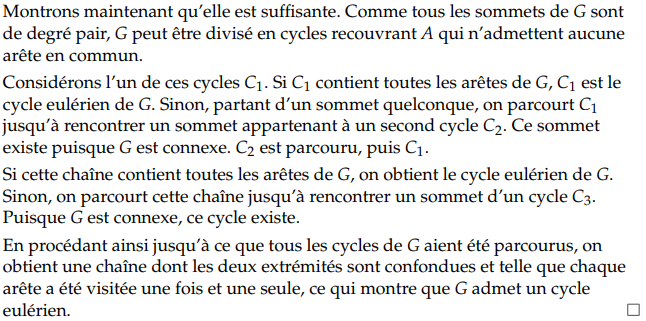
# CHAINES ET CYCLES EULERIENS

Cycles Euleriens

* Theorème : Un graphe non orienté connexe admet un cycle qui traverse chaque arete une et une seule fois ssi chaque sommet a un degré pair. Un tel cycle s’appelle un cycle eulérien.
* Proposition : Soit G un graphe dont tous les sommets sont de degré pair. Alors G peut ëtre divisé en cycles, contenant toutes les arêtes de G, deux quelconques d’entre eux n’ayant aucune arête en commun.

Démonstration :  


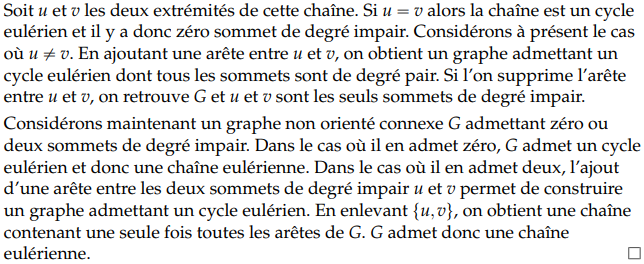
* Théorème : G non orienté connexe admet un cycle eulérien ⬄ ∀s ∈ S, d(s) = 0 [2].

Démonstration :  
  


(Algorithme de Hierholzer : démonstration sur diapo)

Théorème de la chaine eulérienne :  
- Une graphe non orienté connexe admet une chaine qui traverse chaque arete une et une seule fois si et seulement s’il admet zéro ou deux sommets de degré impair. Une telle chaîne s’appelle une chaîne eulérienne.

* G non orienté connexe admet une chaîne eulérienne ⬄0 ou 2 s ∈ S, d(s) = 1 [2]

Démonstration :  


# COLORATION

Coloration de cartes.

* Théorème Il est possible de colorier avec quatre couleurs distinctes n’importe quelle carte découpée en régions connexes, de sorte que deux régions ayant une frontière en commun soient de couleurs distinctes.
* On peut voir ses régions comme des graphes. -> colorier les sommets du graphe. = pas 2 sommet voisins avec la même couleur. = graphe planaire
* Def : un graphe planaire non orienté qu’on peut dessiner sur un plan sans qu’aucune arête n’en croise une autre est un graphe planaire.
* Def : Un graphe non orienté G est planaire ssi il existe une carte ayant G comme graphe associé.
* Algorithme de Welsh-Powell:  
  - Trier les sommets du graphe par ordre décroissant de leur degré  
  - On choisit une couleur (bleu sur l’exemple)  
  - Tant qu’il y a encore des sommets en noir faire  
  - Parcourir la liste triée de sommets en noir et colorier les sommets qui ne sont pas connectés à d’autres sommets de la couleur en cours  
  - Choisir une nouvelle couleur  
   