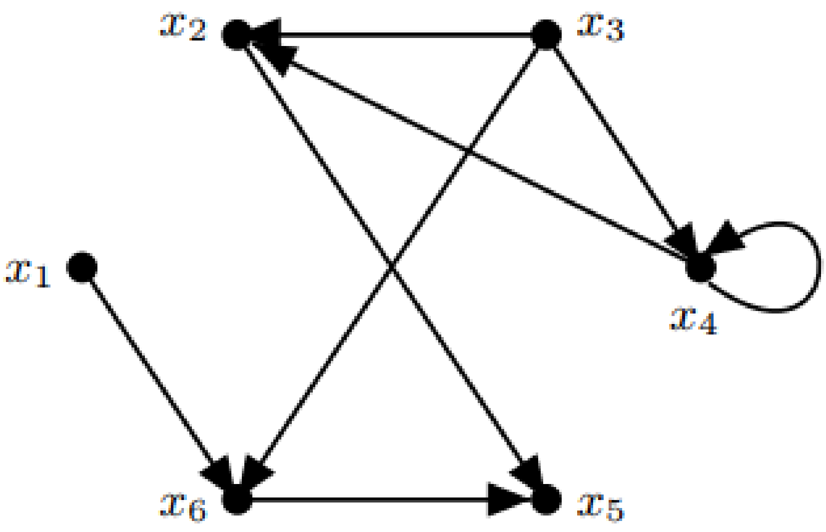
Graphe orienté I

Un graphe orienté $G = (X, U)$ est défini par les deux ensembles:

- $X = \{x_1, x_2, \ldots, x_n\}$ est l'ensemble fini de **sommets**, $n \geqslant 1$.
- $U = \{u_1, u_2, \ldots, u_m\}$ est l'ensemble fini d'**arcs**.
  Chaque élément $u_i \in U$ est une paire ordonnée de sommets, $u_i = (x, y)$. x est appelé **extrémité initiale** de u_i et y est **extrémité terminale**.
  U peut être vide.

Graphe orienté II



$$X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$$

$$U = \{(x_1, x_6), (x_2, x_5), (x_3, x_2), (x_3, x_4), (x_3, x_6), (x_4, x_2), (x_4, x_4), (x_6, x_5)\}$$

Degré d’un sommet I

Soit $G = (X, U)$ *un graphe orienté.*

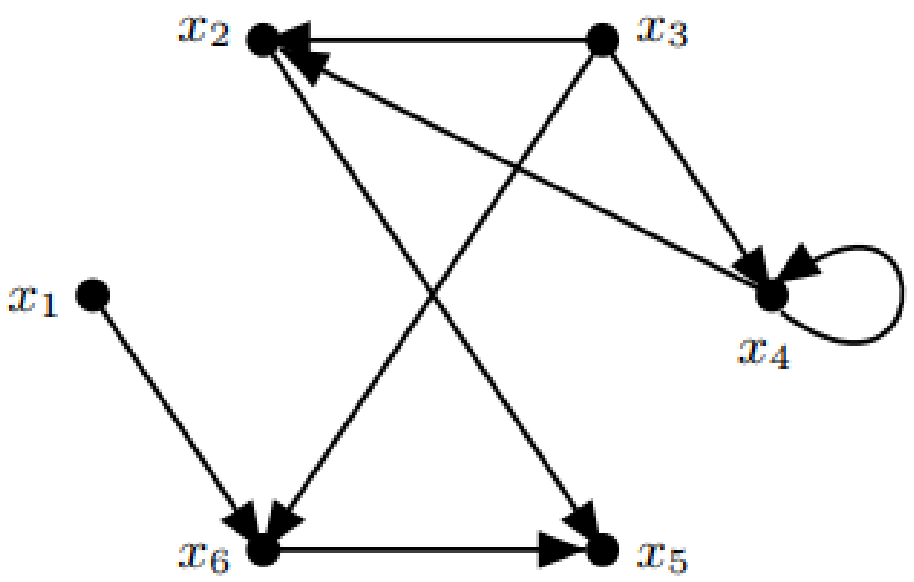
Le demi-degré extérieur d’un sommet $x \in X$, *noté* $d_G^+(x)$, *est égal au nombre d’arcs dont* x *est l’extrémité initiale.*

Le demi-degré intérieur d’un sommet $x \in X$, *noté* $d_G^-(x)$, *est égal au nombre d’arcs dont* x *est l’extrémité terminale.*

Le degré d’un sommet x: $d_G(x) = d_G^+(x) + d_G^-(x)$

Degré d’un sommet II

$$d_G^+(x_1) = 1,\ d_G^-(x_1) = 0,$$
$$d_G^+(x_2) = 1,\ d_G^-(x_2) = 2,$$
$$d_G^+(x_4) = 2,\ d_G^-(x_4) = 2.$$

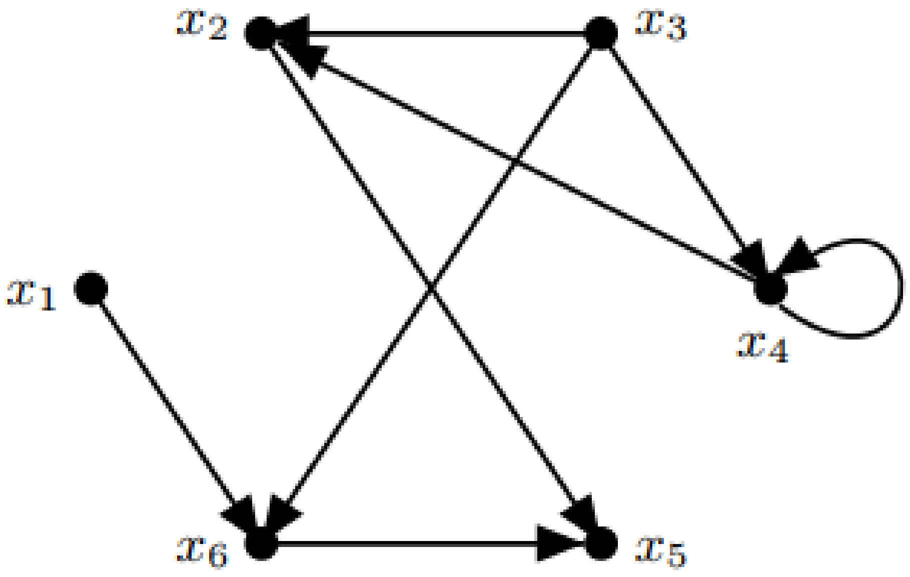


Degré d'un sommet III

Définition (Source. Puits)**.** Soit $G = (V, A)$ graphe orienté et $v \in V$:

- v est une **source (de G)** si et seulement si $\boldsymbol{d^-(v)} = 0$.
- v est un **puits (de G)** si et seulement si $\boldsymbol{d^+(v)} = 0$.

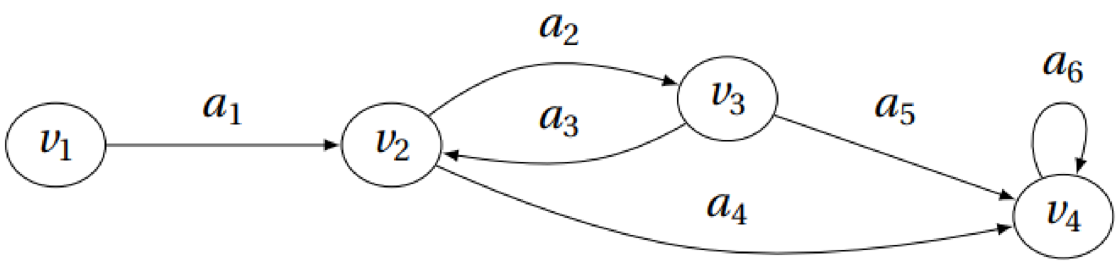
Lemme (Existence de source et puits)**.** Tout graphe orienté sans circuit admet une source et un puits.



Degré d'un sommet IV

Théorème (Lemme des poignées de main).
Soit $G=(V,A)$ un graphe orienté. Alors $\displaystyle\sum_{v\in V} d^-(v)=\sum_{v\in V} d^+(v)=\operatorname{card} A$.

Exemple.



v	v_1	v_2	v_3	v_4
$d^-(v)$	0	2	1	3
$d^+(v)$	1	2	2	1
$d(v)$	1	4	3	4

Vérifions le lemme des poignées de main :

— $\displaystyle\sum_{v\in V} d^-(v)=0+2+1+3=6$

— $\displaystyle\sum_{v\in V} d^+(v)=1+2+2+1=6$

— $\operatorname{card} A=6$

Notion de chemin

On appelle chemin dans un graphe orienté $G = (X, U)$, une 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 tout $1 \leqslant i \leqslant k$, le sommet x_i est extrémité initiale de l'arc u_i et le sommet x_{i+1} est son extrémité terminale.
On dit que γ est un chemin de x_0 vers x_k de longueur k.

$C_2 = x_3 u_6 x_2 u_5 x_5 u_3 x_6$ est une chemin de longueur 3
$C_3 = x_3 u_7 x_4 u_9 x_4 u_8 x_2 u_2 x_6$ est une chemin de longueur 4
$C_4 = x_1 u_1 x_6 u_2 x_2 u_6 x_3 u_7 x_4$ est une chaine de longueur 4

Chaine vs Chemin

Chaîne / Chemin simple

On dit qu'un chemin ou une chaîne est simple si tous les arcs ou les arêtes les composants sont distincts.

Chaîne / Chemin élémentaire

On dit qu'un chemin ou une chaîne est élémentaire si tous les sommets les composants sont distincts.

Notion de Circuit

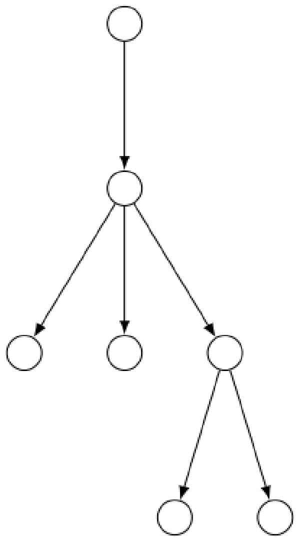
On appelle circuit dans un graphe orienté G=(X, U), tout chemin fermé simple : $\gamma = x_0 u_1 x_1 \ldots x_{k-1} u_k x_k$ tel que $k > 0$, et $x_0 = x_k$. On dit que γ est un circuit de longueur k.

DÉFINITION *On dit qu'un circuit est élémentaire si tous les sommets qui les composent sont distincts.*

Arborescence

Définition (Racine d'un graphe orienté)**.** Soit $G = (V, A, \gamma)$ un graphe orienté et $r \in V$.
r est une **racine de G** si et seulement si il existe un chemin de r à tout sommet de G.

Définition (Arborescence)**.** Une **arborescence** est un graphe orienté admettant une racine et tel que son graphe non orienté associé soit un arbre.



Parcours d'une arborescence I

On présente l'algorithme de parcours en profondeur d'abord ou *Depth-First Search* (DFS) pour une arborescence.

procedure DFS(T, v_0)
préconditions : $T = (V, A, \gamma)$ est une arborescence et $v_0 \in V$
  pour v successeur de v_0 **faire**
    /* *Prévisite de v* */
    DFS(T, v)
    /* *Postvisite de v* */
  fin pour
fin procedure

Algorithme : Parcours en profondeur d'abord d'une arborescence

Proposition (Complexité)**.** Soit $G = (V, A, \gamma)$ une arborescence. Alors la complexité de l'algorithme est $O(\mathrm{card}\,A) = O(\mathrm{card}\,V)$.

Remarque. G est une arborescence, donc $\mathrm{card}\,A = \mathrm{card}\,V - 1$.

Parcours d’un graphe orienté I

On présente l’algorithme de parcours en profondeur d’abord pour un graphe orienté quelconque.

procedure DFS(G, v_0)
préconditions : $G = (V, A, \gamma)$ graphe orienté et $v_0 \in V$
  `visited` : Tableau associatif de<Sommet> de booléen
  pour $v \in V$ **faire**
    `visited`(v) $\leftarrow$ **FAUX**
  fin pour
  / Prévisite de v_0 */*
  `visited`(v_0) $\leftarrow$ **VRAI**
  pour v successeur de v_0 **faire**
    si `visited`(v) **alors**
      / Revisite de v */*
    sinon
      / Prévisite de v */*
      DFS(T, v)
      / Postvisite de v */*
    fin si
  fin pour
  / Postvisite de v_0 */*
fin procedure

Algorithme : Parcours en profondeur d’abord d’un graphe orienté

Parcours DFS : complexité

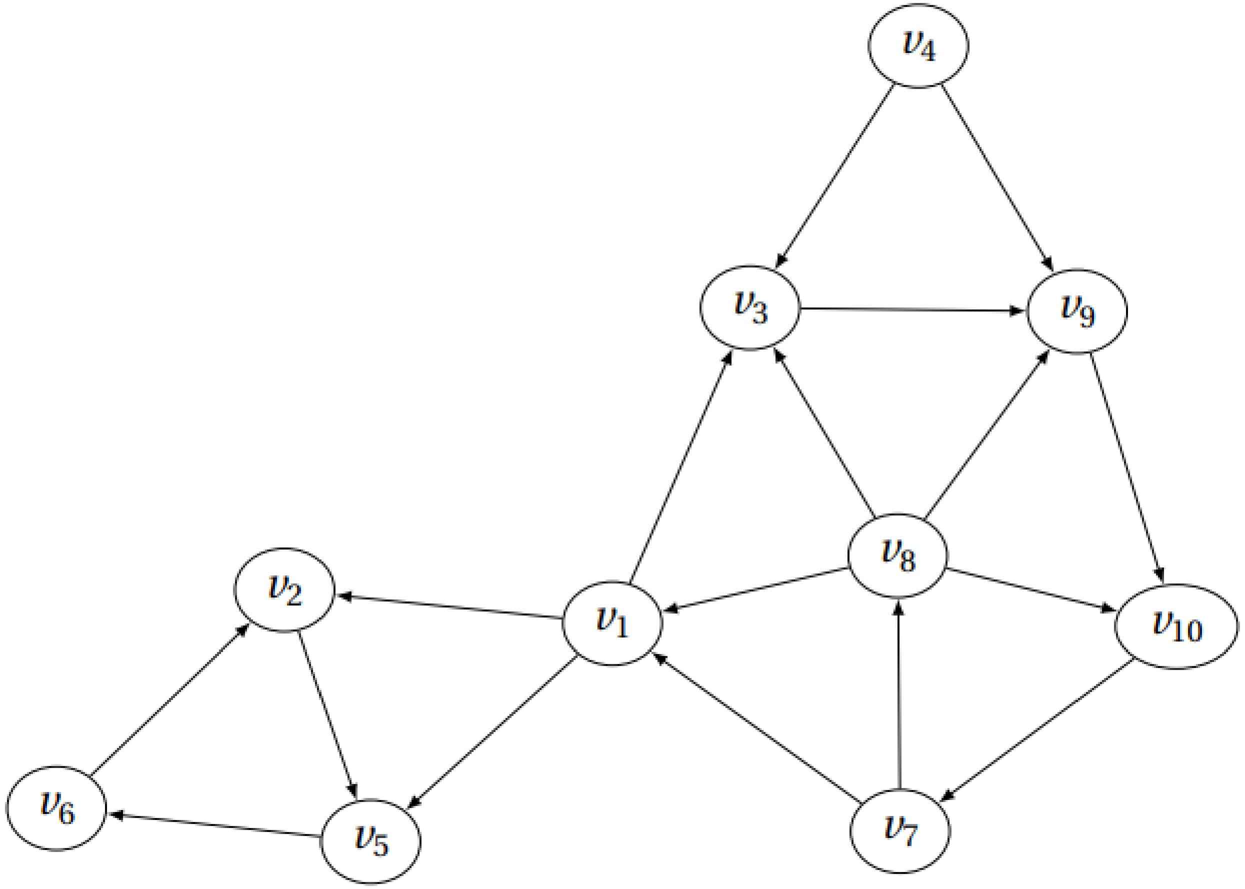
Proposition (Complexité). Soit $G = (V, A, \gamma)$ un graphe orienté.
Alors la complexité de l'algorithme est $\mathrm{O}(\operatorname{card} A + \operatorname{card} V)$.

Remarque. L'expression de la complexité se décompose en deux parties :

— le parcours des sommets : $\mathrm{O}(\operatorname{card} A)$;

— le traitement de `visited` : $\mathrm{O}(\operatorname{card} V)$.

Parcours d’un graphe orienté II



Forte Connexité

DÉFINITION *Un graphe orienté $G=(X, U)$ est fortement connexe s'il existe entre chaque paire de sommets x et $y \in X$ $(x \neq y)$ un chemin de x à y et un chemin de y à x.*

Composante fortement connexe

Composantes fortement connexes (cfc)

Soit un graphe $G = (X, U)$:

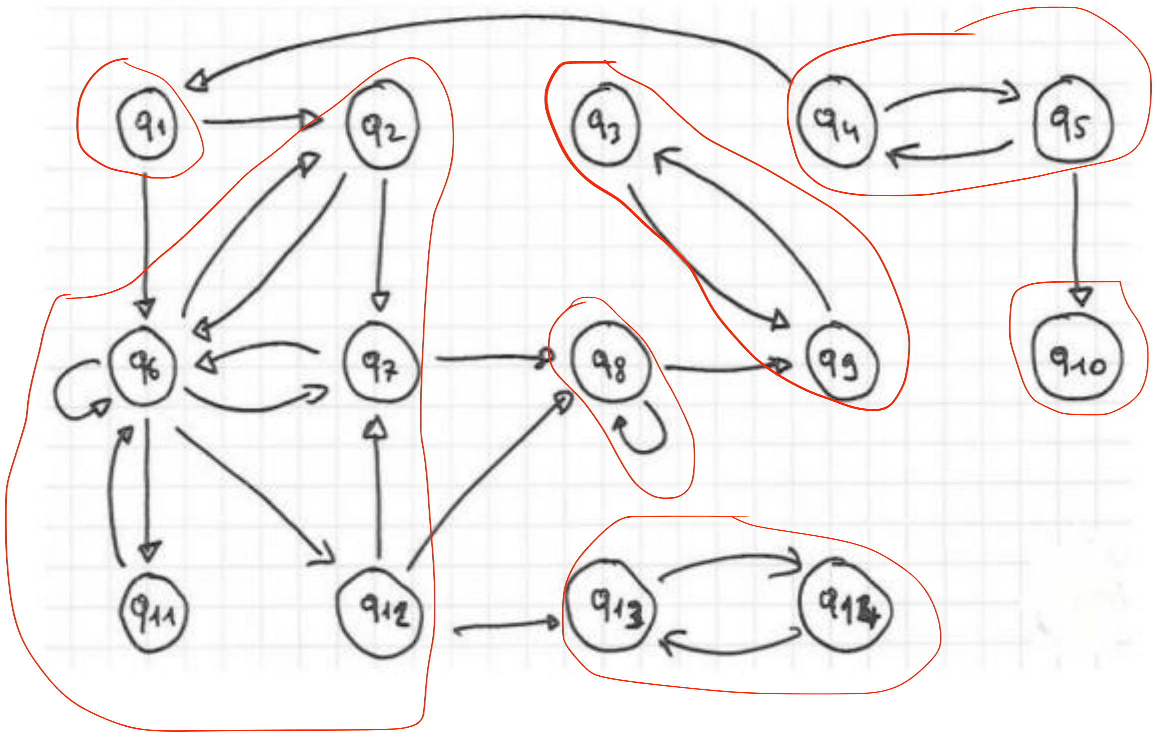
La relation

$$xRy \Leftrightarrow \begin{cases} soit\ x = y \\ soit\ \exists\ un\ chemin\ de\ x\ vers\ y\ et\ un\ chemin\ de\ y\ vers\ x \end{cases}$$

est une relation d'équivalence, les classes d'équivalence $X_1, X_2, \ldots X_k$ induites par R sur l'ensemble de sommets X engendre les composantes fortement connexes de G.
G comporte alors k composantes fortement connexes.

Un graphe qui contient une seule composante fortement connexe est dit graphe fortement connexe.

Composante fortement connexe



Composante fortement connexe

Application

Faites un état de l’art des algorithmes pour la détection des composantes fortement connexes d’un graphe orienté.

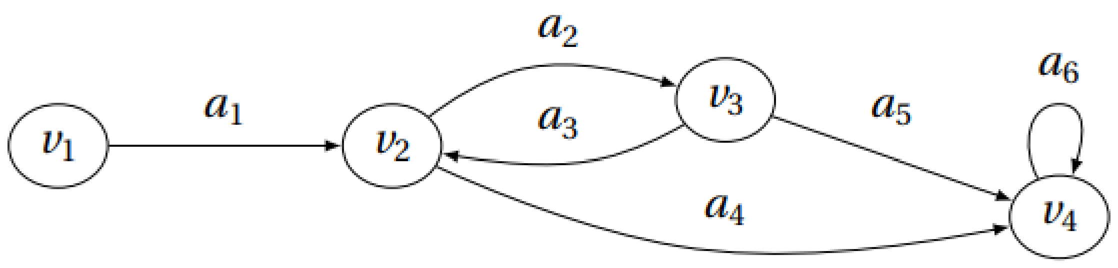
Tri topologique

Définition (Tri topologique). Soit $G = (V, A, \gamma)$ un graphe orienté.
Un **tri topologique de G** est une bijection ϕ de $V \to [\![1, \mathrm{card}\, V]\!]$ telle que :

$$\forall a \in A \text{ d'origine } v \text{ et destination } v',\ \phi(v) < \phi(v')$$

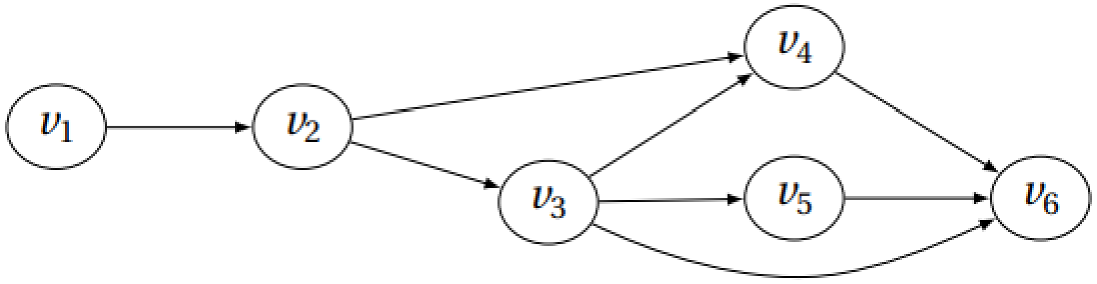
Proposition (Existence de tri topologique). Un graphe orienté admet un tri topologique si et seulement si il n'admet pas de circuit.

Tri topologique



?

Tri topologique



?

Tri topologique : application

Déterminer (si il existe) un tri topologique du graphe suivant :