# Graphes

IMIE – 13 & 15 décembre 2017

# Plan

- Un graphe est constitué de deux ensembles :
  - un ensemble de « points » qu'on appelle **sommets**
  - un ensemble de liens entre ces points

Par exemple, on peut représenter les « connexions » entre les utilisateurs d'un réseau social par un graphe :

- les sommets seront les identifiants des différents utilisateurs,
- et les arêtes seront les connexions entre les utilisateurs.

On peut encore modéliser un plan de réseau de bus avec un graphe :

- les sommets seront les différentes stations,
- et les arêtes seront les stations qui se suivent sur une des lignes de bus.

# Deux types de graphes

Il existe deux types de graphes

- Un graphe **non orienté** est défini par un couple $(S, A)$ de sommets et d'arêtes
- Si deux sommets $x$ et $y$ sont reliés :
  - on écrit $(x, y) \in A$
  - on dit que $x$ et $y$ sont **adjacents**

Par exemple, si le graphe représente les liens d'amitié sur Facebook, on peut utiliser un graphe non orienté :

- si $x$ est ami avec $y$ alors l'inverse est vrai, le lien va dans les deux sens

# Deux types de graphes

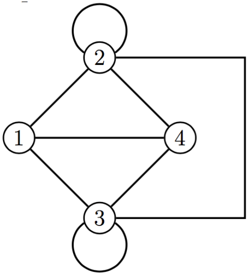
- Un graphe **orienté** est défini par un couple $(S, A)$ de sommets et d'arcs
- Si deux sommets $x$ et $y$ sont reliés par un lien de $x$ vers $y$ :
  - on écrit $(x, y) \in A$
  - on dit que $x$ est **prédécesseur** de $y$
  - et $y$ est **successeur** de $x$

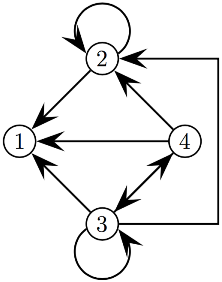
Par exemple, si le graphe représente les liens sur Twitter, on peut utiliser un graphe orienté :

- si $x$ « suit » $y$ sur Twitter, l'inverse n'est nécessairement vrai, les liens ne vont que dans un seul sens
- si $x$ « suit » $y$ et $y$ « suit » $x$, on aura deux arcs de sens inverses entre $x$ et $y$

## Exercice

- Pour les deux graphes ci-dessous, l'ensemble $S$ de sommets est identique : $\{1, 2, 3, 4\}$
- En revanche, l'ensemble $A$ d'arêtes du graphe non orienté est différent de l'ensemble $A'$ d'arcs du graphe orienté : faire la liste des couples de sommets contenus dans ces deux ensembles

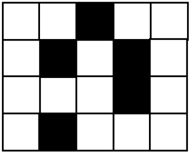


## Exercice

On représente un labyrinthe sous la forme d'un tableau de dimensions $(m, n)$ où les cases sont remplies par des espaces ou des murs

- On veut transformer ce labyrinthe en un graphe. Pour cela, on donne un numéro à chaque case du tableau, ce qui définit les sommets du graphe. Il y aura un arc entre deux états si l'on peut passer d'une case à l'autre. Construire le graphe associé au labyrinthe de la figure.



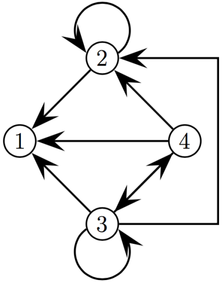
# Implémentation des graphes

- Différentes implémentations : matrice d'adjacence, matrice d'incidence, liste d'adjacence
- L'implémentation va avoir un impact sur le coût de certaines opérations :
  - recherche de l'existence d'une arête entre deux sommets
  - trouver les « voisins » d'un sommet

# Matrice d'adjacence

- Une matrice d'adjacence est une matrice **booléenne** (elle ne contient que des 0 et des 1) qui indique si deux sommets sont reliés dans le graphe
- $M(i,j) = 1 \Leftrightarrow (i,j) \in A$
- Matrice indicée sur $S \times S$
  - nombre de lignes = nombre de sommets
  - nombre de colonnes = nombre de sommets
- Attention : pour pouvoir écrire la matrice d'adjacence il faut avoir « ordonné » les sommets. Le choix de l'ordre est arbitraire, mais il n'est pas indiqué dans la matrice d'adjacence !

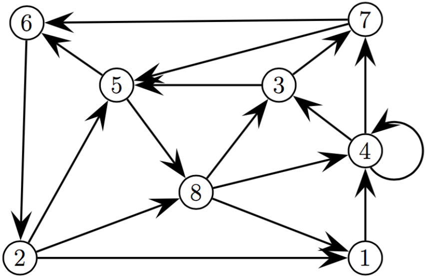
# Matrice d'adjacence




Matrice d'adjacence : $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

- Lignes : successeurs
- Colonnes : prédécesseurs

# Exercices

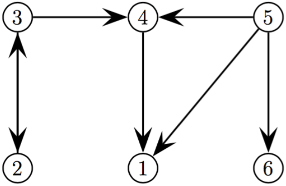



- Écrire la matrice d'adjacence associée au graphe ci-dessus.
- Écrire un algorithme qui construit la matrice d'adjacence associée à un graphe orienté. L'algorithme prend en entrée les ensembles $S$ et $A$ (respectivement les sommets et les arcs).

# Matrice d'incidence

- Matrice indicée sur $S \times A$ (ensemble de sommets, ensemble d'arcs)
  - nombre de lignes = nombre de sommets
  - nombre de colonnes = nombre d'arcs
- $$M(x,a) = \begin{cases} 1 & \text{si } x \text{ est l'origine de l'arc } a \\ \text{-1} & \text{si } x \text{ est l'extrémité de l'arc } a \\ 0 & \text{sinon} \end{cases}$$

# Matrice d'incidence

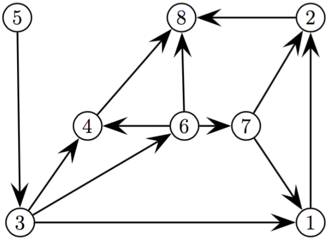



Matrice d'incidence :

$$\begin{bmatrix} 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

- Ensemble des arcs : $\{(2,3),(3,2),(3,4),(4,1),(5,1),(5,4),(5,6)\}$

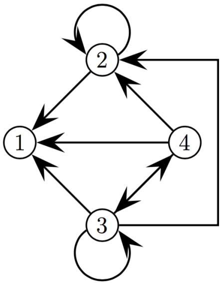
# Exercices




- Lister les arcs (couples de sommets) contenus dans le graphe ci-dessus.
- Écrire la matrice d'incidence associée à ce graphe.
- Écrire un algorithme qui construit la matrice d'incidence associée à un graphe orienté. L'algorithme prend en entrée les ensembles $S$ et $A$ (respectivement les sommets et les arcs).

# Liste d'adjacence

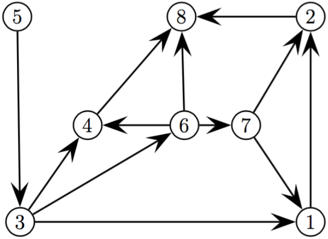
- La liste d'adjacence contient, pour chaque sommet $x$ :
  - la liste des sommets $y$ tels que $(x, y) \in A$




Liste d'adjacence :

[ [ ], [1,2], [1,2,3,4], [1,2,3] ]

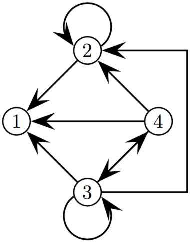
# Exercices




- Écrire la liste d'adjacence associée au graphe ci-dessus.
- Écrire un algorithme qui construit la liste d'adjacence associée à un graphe orienté. L'algorithme prend en entrée les ensembles $S$ et $A$ (respectivement les sommets et les arcs).

# Exercices

- Écrire deux algorithmes qui déterminent si il **existe un arc entre deux sommets** $s_1$ et $s_2$ d'un graphe. Le premier s'appuie sur une représentation du graphe par matrice d'adjacence, et le second sur une représentation par liste d'adjacence.
- Comparer le coût de l'opération si l'on cherche à déterminer l'existence d'un arc entre $s_1 = 3$ et $s_2 = 4$ pour les deux implémentations (matrice d'adjacence, liste d'adjacence) du graphe ci-dessous.

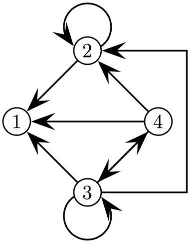



$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

[ [ ], [1,2], [1,2,3,4], [1,2,3] ]

# Exercices

- Écrire deux algorithmes qui renvoient tous les **sommets voisins** d'un sommet $s_1$ d'un graphe. Le premier s'appuie sur une représentation du graphe par matrice d'adjacence, et le second sur une représentation par liste d'adjacence.
- Comparer le coût de l'opération si l'on cherche à tous les sommets voisins du sommet $s = 1$ pour les deux implémentations (matrice d'adjacence, liste d'adjacence) du graphe ci-dessous.

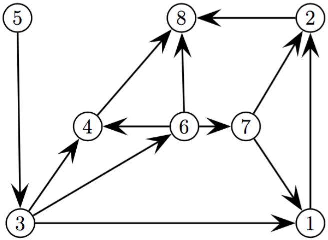



$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

$[\ [\ ],\ [1,2],\ [1,2,3,4],\ [1,2,3]\ ]$

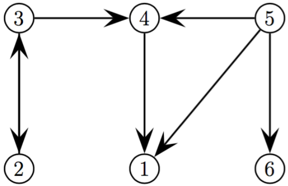
- Chemin de longueur $p$, d'origine $x$ et d'extrémité $y$ : $C = (s_0, \ldots s_p)$ tel que :
  - $s_0 = x$
  - $s_p = y$
  - $(s_i, s_{i+1}) \in A$ pour $0 \leq$
- On dit qu'un sommet $y$ est **accessible** depuis un sommet $x$ s'il existe un chemin qui va de $x$ vers $y$
- Un graphe non orienté est dit **connexe** si quels que soient les sommets $x$ et $y$, il existe un chemin d'origine $x$ et d'extrémité $y$
- Un graphe orienté est dit :
  - **connexe** si en oubliant l'orientation des arêtes, le graphe est connexe
  - **fortement connexe** s'il existe un chemin orienté depuis tout sommet $x$ vers tout sommet $y$

# Exercices



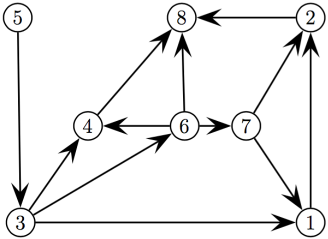

- Quels sont les sommets accessibles depuis le sommet 7 ? Depuis le sommet 5 ?
- Le graphe ci-dessus est-il connexe ?
- Est-il fortement connexe ? Pourquoi ?

# Exercices




- Écrire la matrice d'adjacence $M$
- Compléter le graphe avec les chemins de longueur 2 (ne pas oublier les chemins allant d'un sommet vers lui-même)
- Écrire la matrice d'adjacence $M'$ du graphe résultant
- Calculer le produit $M^2 = M \times M$
- Calculer $M + M^2$ ( ! $M$ est une matrice booléenne on fait l'union) : à quoi cette matrice est-elle égale ?

- Fermeture transitive : on explicite tous les liens entre sommets même ceux qui ne sont pas des liens directs



## Calcul de la fermeture transitive à l'aide de la matrice d'adjacence $M$

- Étape 1 : On pose $S_1 = M$
- Étape 2 :
  - On calcule $M^2$
  - On pose $S_2 = S_1 \cup M^2$ (c'est-à-dire $S_2 = M \cup M^2$)
  - Si $S_2 = S_1$ on renvoie $S_1$ (c'est la matrice d'adjacence de la fermeture transitive)
- Étape 3 :
  - On calcule $M^3$
  - On pose $S_3 = S_2 \cup M^3$ (c'est-à-dire $S_3 = M \cup M^2 \cup M^3$)
  - Si $S_3 = S_2$ on renvoie $S_2$ (c'est la matrice d'adjacence de la fermeture transitive)
- On réitère ces opérations tant que $S_i \neq S_{i-1}$

- Écrire un algorithme qui calcule la fermeture transitive d'un graphe à partir de sa matrice d'adjacence