Outils pour la conception des algorithmes.

Johanne Cohen[1]

Chercheuse, LISN-CNRS.
email : johanne.cohen@lisn.fr
https://www.lri.fr/∼jcohen/

Comment résoudre un problème

- Modéliser le problème sous forme d'outils informatiques
- Conception des algorithmes :
  - Reconnaitre l'algorithme déjà existant qui s'applique
  - Inventer de nouveaux algorithmes
- Analyse :
  - Comprendre ce que retourne l'algorithme
  - Comprendre la rapidité de l'algorithme
  - Comprendre la qualité de la solution retournée de l'algorithme

Termes abordés

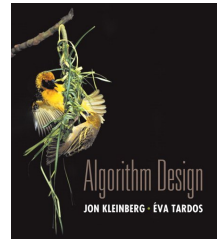
- Outils informatiques :
  1. listes, tableaux, ...
  2. les graphes, les arbres

- Méthodes d'algorithmes :
  1. Les algorithmes gloutons et la programmation dynamique
  2. Algorithmes d'approximation
  3. Algorithmes probabilistes.
  4. Algorithmes Online.

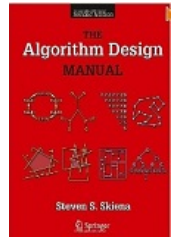
Evaluation : un contrôle continu, un projet et un examen final.

Références du cours

1. *Algorithm Design*, Jon Kleinberg, Eva Tardos, Addison-Wesley, 2006.



2. *The Algorithm Design Manual*, Steven Skiena, Springer 2014.



Plan

Les graphes : un outil pour la modélisation

Terminologie sur les graphes

Les graphes classiques

- Les graphes planaires
- Graphes bipartis
- Les graphes complets
- Les arbres
- Structure de graphes

Représentation des graphes

- Matrice d'adjacences
- Liste de successeurs

Affectation d'étudiants dans les hôpitaux

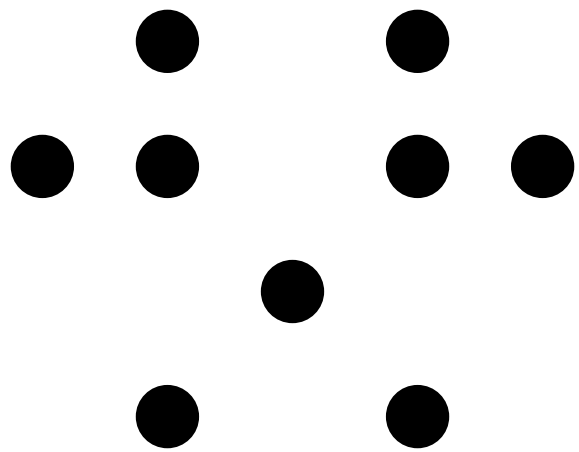
- Description & Formalisation
- Quel type d'objects il faut calculer ?
- Extension : et si les acteurs avaient des préférences

Conclusion

Les graphes : un outil pour modéliser les relations entre entités

Un graphe est donné par une paire $G = (V, E)$, où :

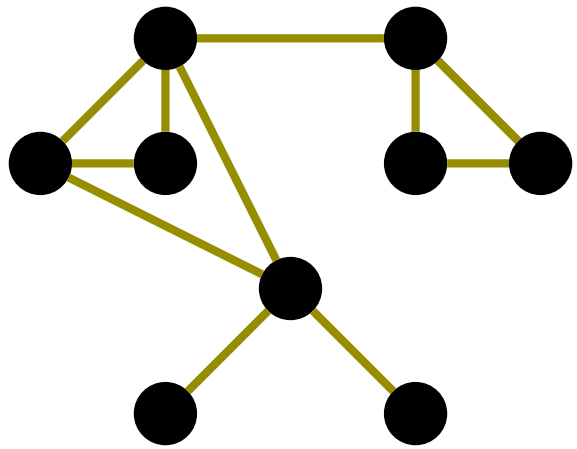
- V est un ensemble de sommets (entités)



Les graphes : un outil pour modéliser les relations entre entités

Un graphe est donné par une paire $G = (V, E)$, où :

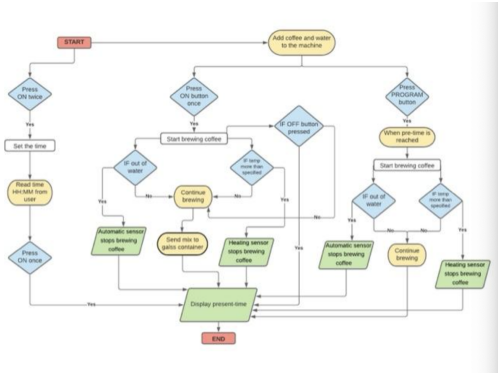
- V est un ensemble de sommets (entités)
- $E \subset V \times V$ est un ensemble d'arêtes u, v avec $u, v \in V$. (relations entre entités)



Exemple : Applications des graphes

Beaucoup de problèmes se modélisent par des objets et des relations entre objets.

Graphes d'événements /Workflows

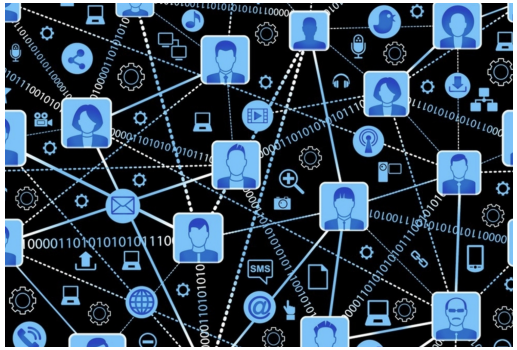


source : https ://www.chegg.com/

Réseaux de transport

source : https ://www.ratp.fr

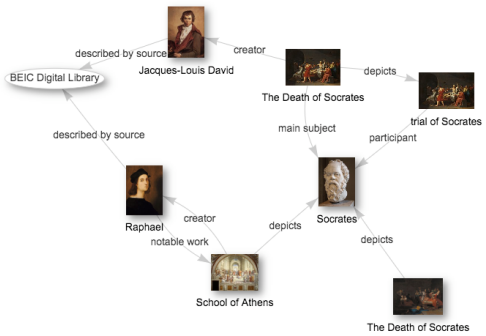
Réseaux sociaux



source : https ://news.mit.edu

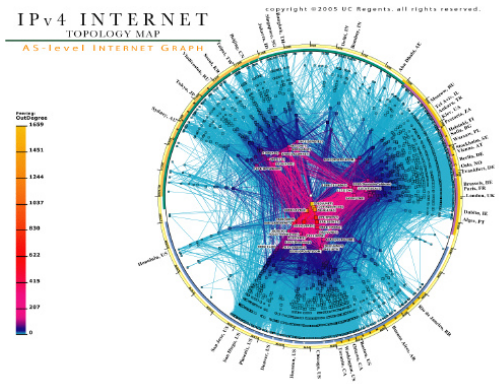
Exemple : Applications des graphes

Graphes de connaissances connaissances



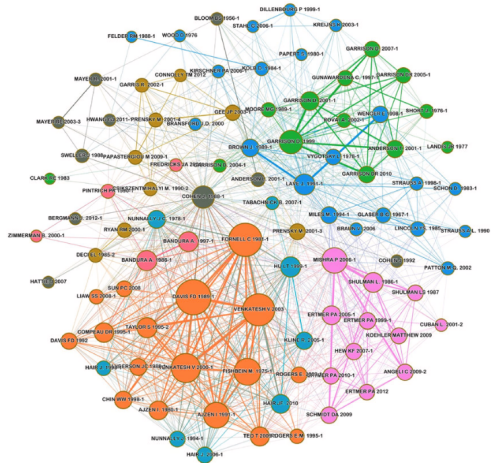
source : https ://www.wikidata.com/

Réseaux de communication



source : projet CADAI

Réseaux de citations



source : https ://rflow.ai/

Autres exemples : Molécules, graphes de code, formes 3D, . . .

Observation : Les graphes du monde réel peuvent être massifs

- Réseaux sociaux : des milliards de nœuds représentant des individus, avec des modèles d'interaction complexes et évolutifs (ex. : Facebook, Twitter).
- Réseaux de communication : une infrastructure composée de milliards d'appareils (smartphones, routeurs, serveurs) interconnectés via des appels, des messages ou des échanges de données.
- Connectivité cérébrale : des réseaux représentant le câblage du cerveau humain, avec environ 10^{11} neurones et 10^{14} synapses, modélisant la connectivité fonctionnelle ou structurelle.

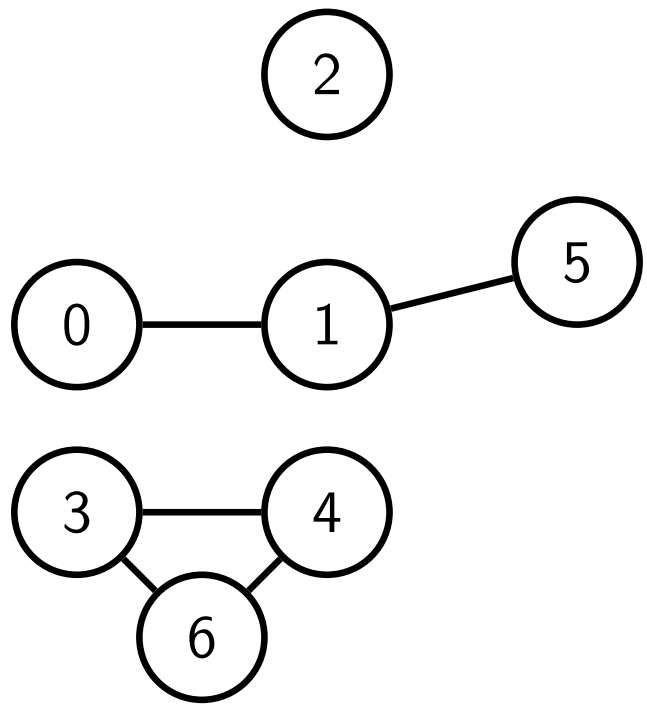
Définition

Un graphe $G = (V, E)$ est constitué de :

- V : Ensemble de sommets (nœuds)
- $E \subseteq V \times V$: Ensemble d'arêtes

Un exemple :

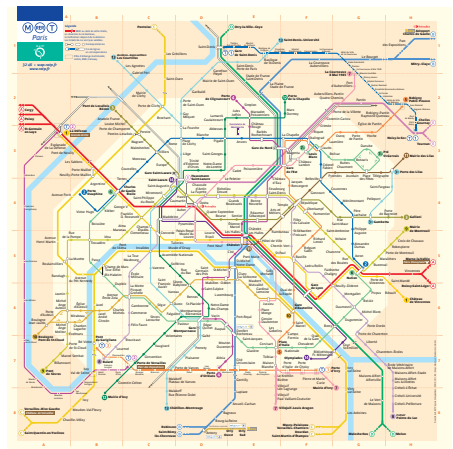
- $V = \{0, 1, 2, 3, 4, 5, 6\}$
- $E = \{\{0, 1\}, \{3, 4\}, \{5, 1\}, \{6, 3\}, \{6, 4\}\}$



Comment définir un graphe ?

- Comment construire un graphe :
  - Les **nœuds** représentent des entités (ex. : stations, utilisateurs, atomes)
  - Les **arêtes** représentent des relations

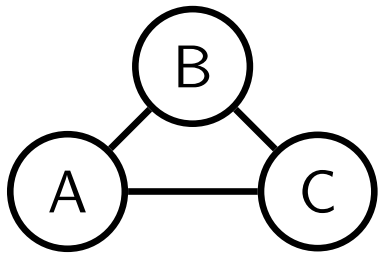
- Exemple : Représentation d'un réseau de transport
  - **Nœuds** : Stations
  - **Arêtes** : Connexions physiques entre les stations
  - **Poids des arêtes** : Temps de trajet, distance, ou coût de correspondance



Graphes non orientés versus graphes orientés

Graphe non orienté

- Les arêtes n'ont pas de direction
- $(u, v) \equiv (v, u)$

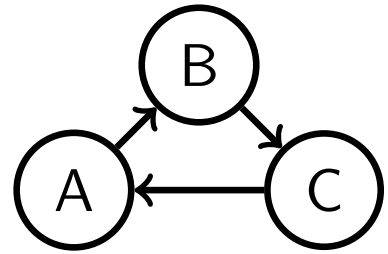


Exemples :

- Réseaux d'amitié
- Co-auteurs
- Interactions protéiques

Graphe orienté (Digraphe)

- Les arêtes ont une direction
- $(u, v) \neq (v, u)$



Exemples :

- Historiques d'appels téléphoniques
- Abonnés sur les médias sociaux
- Liens entre les pages Web
- Flux de trafic

Plan

Les graphes : un outil pour la modélisation

Terminologie sur les graphes

Les graphes classiques

- Les graphes planaires
- Graphes bipartis
- Les graphes complets
- Les arbres
- Structure de graphes

Représentation des graphes

- Matrice d'adjacences
- Liste de successeurs

Affectation d'étudiants dans les hôpitaux

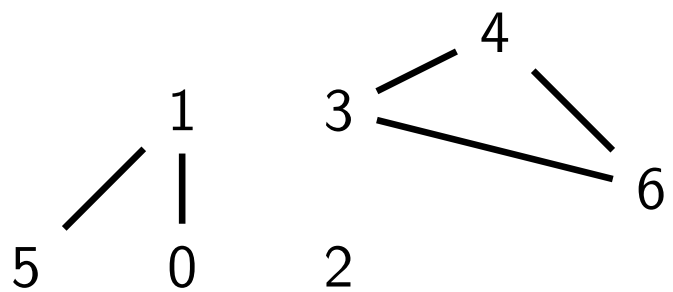
- Description & Formalisation
- Quel type d'objects il faut calculer ?
- Extension : et si les acteurs avaient des préférences

Conclusion

Vocubulaire

- u et v sont dits voisins s'il y a une arête entre u et v.
- Le degré de u est le nombre de voisins de u.
- Un sommet de degré 0 est dit isolé : il n'est relié à aucun autre sommet.

- Remarque : (sauf autre convention explicite)
  - Les boucles ne sont pas autorisées.

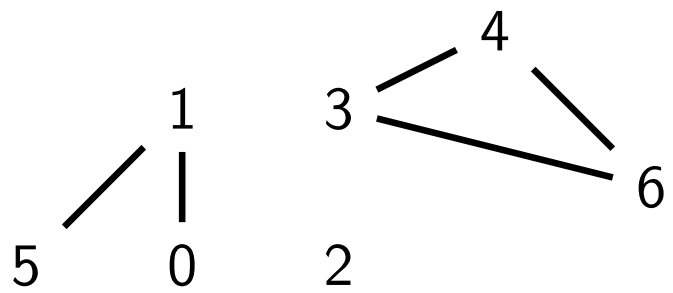


Vocabulaire : chemins et cycles

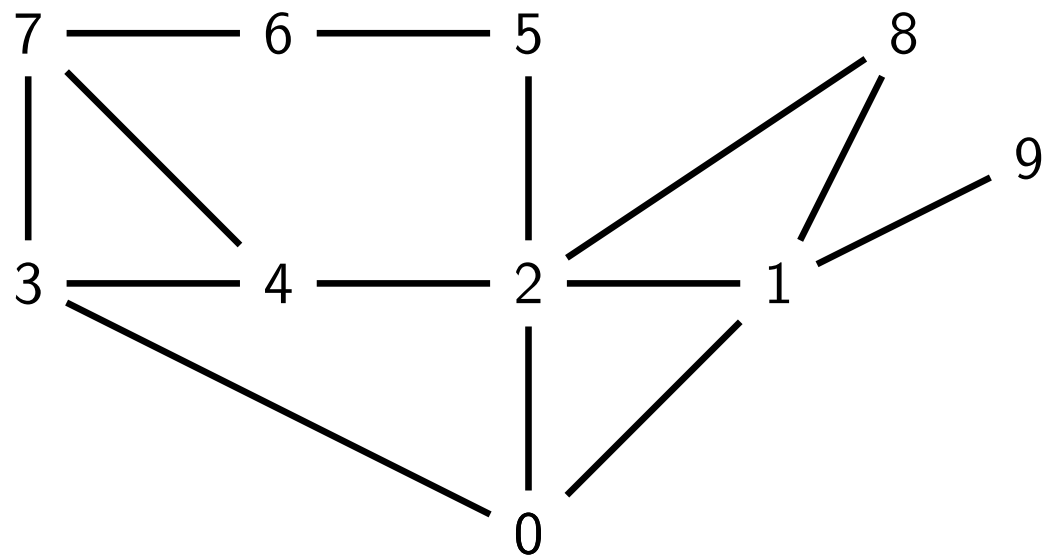
- Un chemin du sommet s vers le sommet t est une suite $e_0, e_1, \cdots, e_n$ de sommets telle que

$$e_0 = s,\ e_n = t,\ (e_{i-1}, e_i) \in E, \text{ pour tout } 1 \leq i \leq n.$$

  - n est appelé la longueur du chemin,
  - on dit que t est joignable à partir de s.
  - Le chemin est dit simple si les e_i sont distincts deux-à-deux.
- Un cycle est un chemin de longueur non-nulle avec $e_0 = e_n$.
- s est dit à distance n de t s'il existe un chemin de longueur n entre s et t, mais aucun chemin de longueur inférieure.

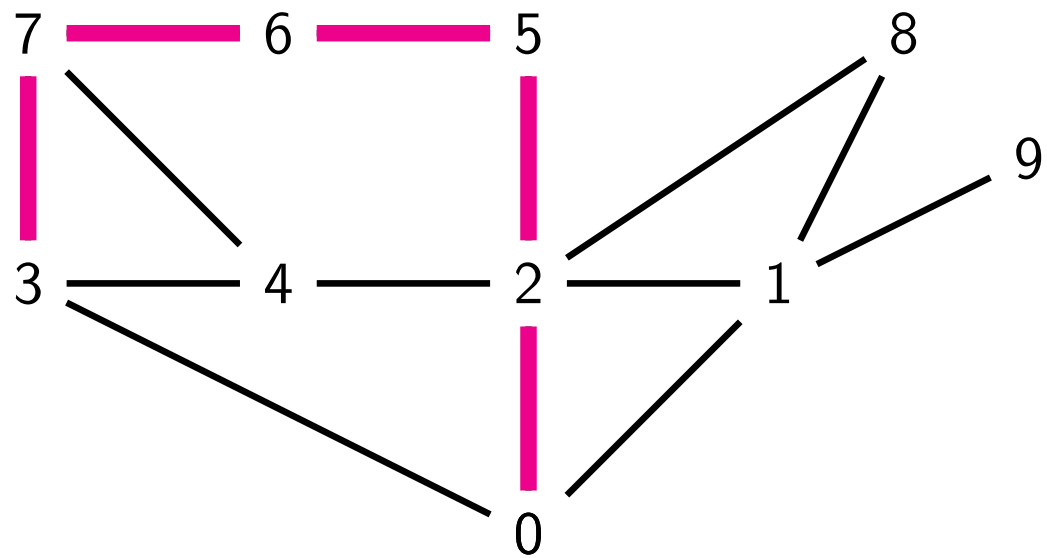


Exemple sur les notions : chemins et de cycles



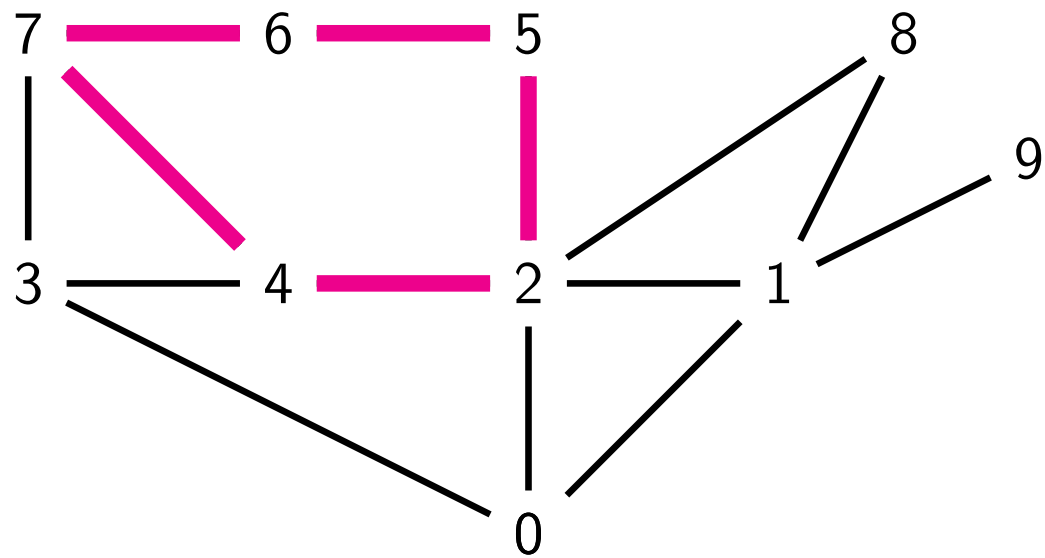
Exemple sur les notions : chemins et de cycles

- Un chemin de 3 à 0 $= \{3, 7, 6, 5, 2, 0\}$



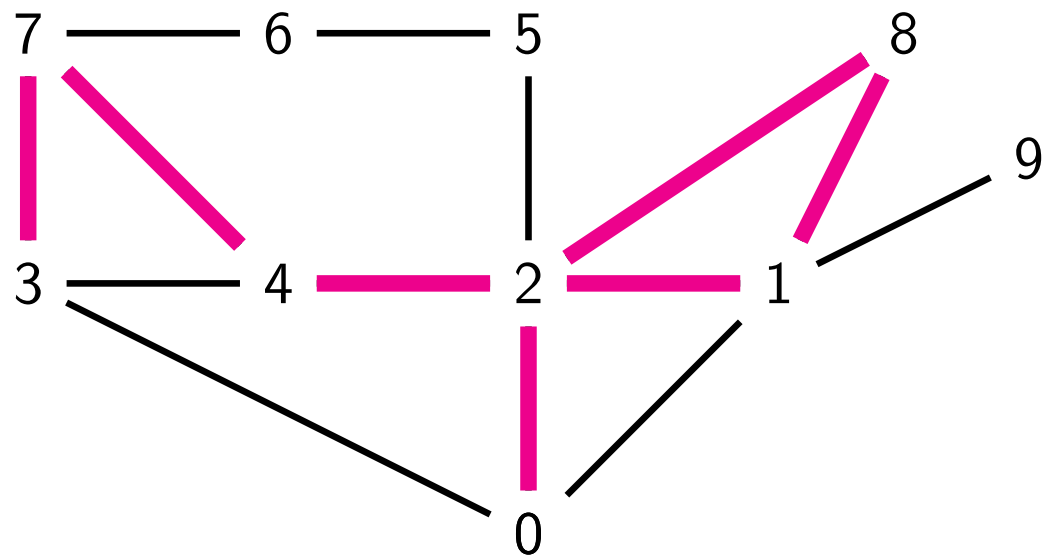
Exemple sur les notions : chemins et de cycles

- Un chemin de 3 à 0 $= \{3, 7, 6, 5, 2, 0\}$
- Un cycle $= \{7, 4, 2, 5, 6, 7\}$



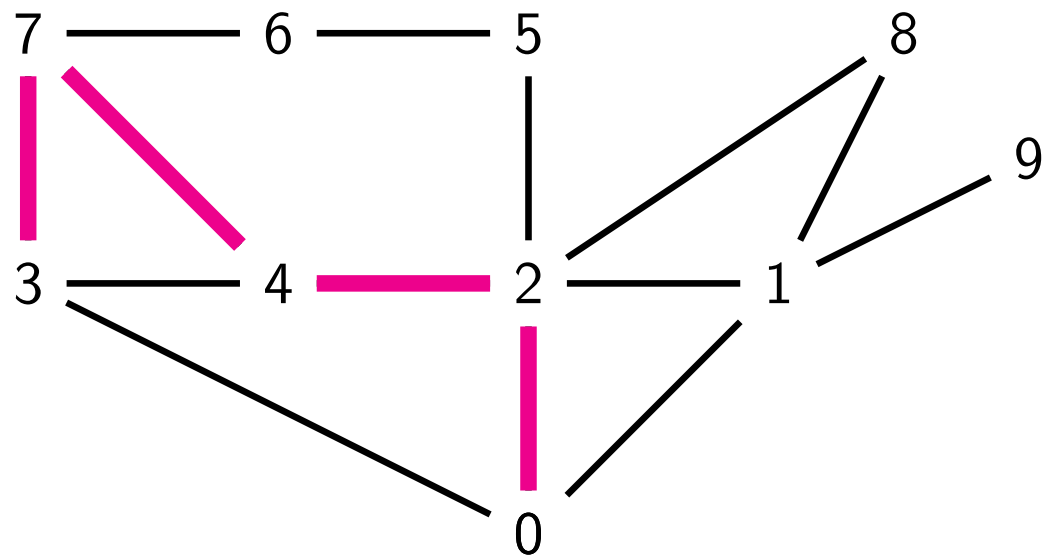
Exemple sur les notions : chemins et de cycles

- Un chemin de 3 à 0 $= \{3, 7, 6, 5, 2, 0\}$
- Un cycle $= \{7, 4, 2, 5, 6, 7\}$
- Un chemin de 3 à 0 $= \{3, 7, 4, 2, 8, 1, 2, 0\}$,

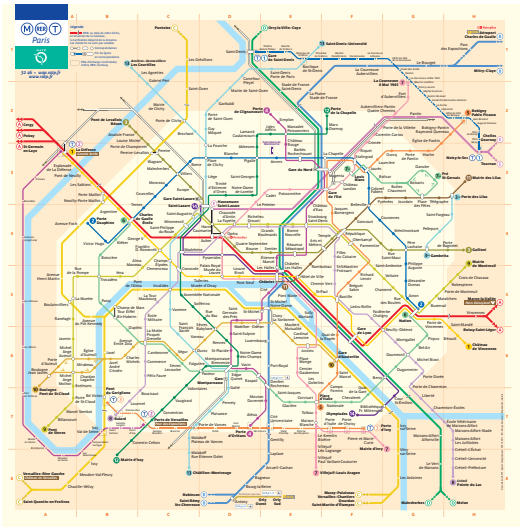


Exemple sur les notions : chemins et de cycles

- Un chemin de 3 à 0 $= \{3, 7, 6, 5, 2, 0\}$
- Un cycle $= \{7, 4, 2, 5, 6, 7\}$
- Un chemin de 3 à 0 $= \{3, 7, 4, 2, 8, 1, 2, 0\}$ ou $\{3, 7, 4, 2, 0\}$,



Exemples de problèmes typiques

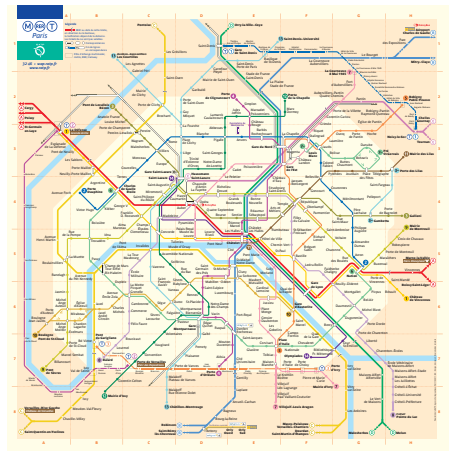


1. Quel est le plus court chemin (en distance ou en temps) pour se rendre d'une ville à une autre ?
2. Comment organiser un planning de taches ?
3. Peut-on mettre une rue en sens unique sans rendre impossible la circulation en ville ?
4. Comment organiser une tournée d'un voyageur de commerce.

Rappel : Représentation d'un réseau de transport

- **Nœuds** : Stations
- **Arêtes** : Connexions physiques entre les stations
- **Poids des arêtes** : Temps de trajet, distance, ou coût de correspondance

Considérations sur le calcul d'itinéraire :



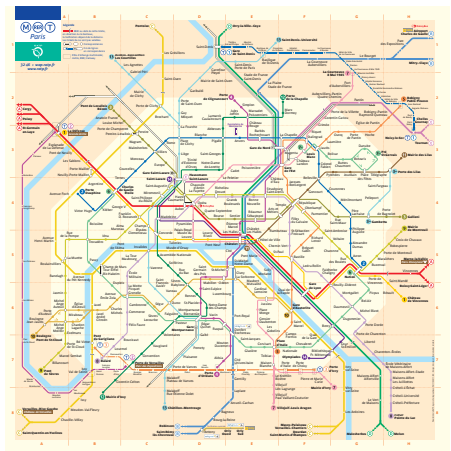
source : ratp.

Rappel : Représentation d'un réseau de transport

- **Nœuds** : Stations
- **Arêtes** : Connexions physiques entre les stations
- **Poids des arêtes** : Temps de trajet, distance, ou coût de correspondance

Considérations sur le calcul d'itinéraire :

- Est-ce que je peux atteindre une station ?



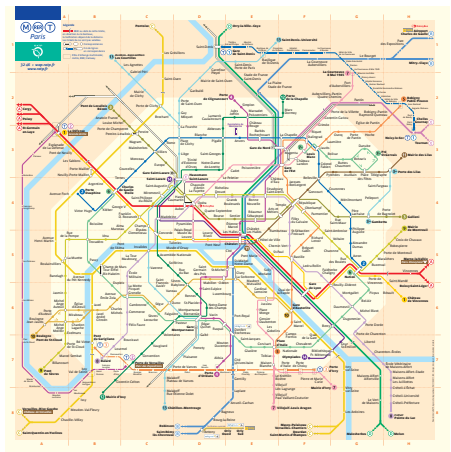
source : ratp.

Rappel : Représentation d'un réseau de transport

- **Nœuds** : Stations
- **Arêtes** : Connexions physiques entre les stations
- **Poids des arêtes** : Temps de trajet, distance, ou coût de correspondance

Considérations sur le calcul d'itinéraire :

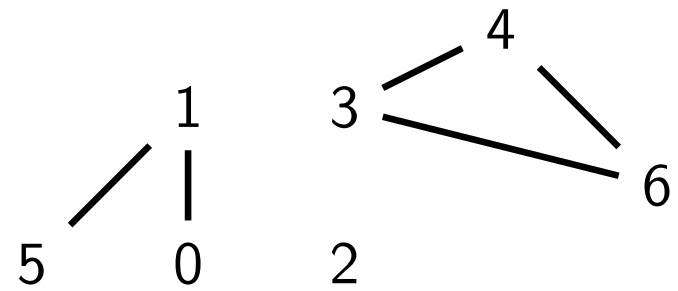
- Est-ce que je peux atteindre une station ?
- Si oui, comment puis-je minimiser le temps de transport ?



source : ratp.

Vocabulaire : composantes connexes

- La relation “être joignable” est une relation d’équivalence.
- Les classes d’équivalence sont appelées les composantes connexes.



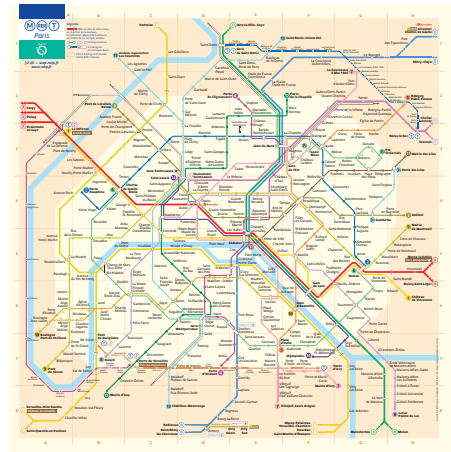
- Un graphe est dit connexe s’il n’y a qu’une seule classe d’équivalence.
  - Autrement dit, tout sommet est joignable à partir de tout sommet.

Retour sur l’exemple du temps tansport

Répondre à la question "Est-ce que je peux atteindre une station ?" revient à déterminer si la station de départ et la station d’arrivée appartiennent à la même composante connexe du réseau.

Rappel : Représentation d'un réseau de transport

- **Nœuds** : Stations
- **Arêtes** : Connexions physiques entre les stations
- **Poids des arêtes** : Temps de trajet, distance, ou coût de correspondance



Considérations sur le calcul d'itinéraire :

comment puis-je minimiser le temps de transport ?

La fonction objectif varie selon le cas d'usage :

- On peut minimiser : le temps de trajet, les correspondances, ou la distance
- On peut inclure des contraintes : accessibilité, capacité

Le problème du plus court chemin

Fait connu

Calculer un plus court chemin dans un graphe pondéré (les poids sont sur les arêtes) peuvent se faire en temps polynomial $\mathcal{O}(|E| + |V|)$.

En utilisant l'algorithme précédent, peut-on calculer un chemin entre 2 stations de métro sachant que

1. l'usager veut minimiser son temps de transport
2. ou l'usager veut minimiser le nombre de changements ;
3. ou l'usager veut minimiser le temps consacré aux chanchements ;
4. ou l'usager veut éviter de changer à une station fixée

Plan

Les graphes : un outil pour la modélisation

Terminologie sur les graphes

Les graphes classiques
- Les graphes planaires
- Graphes bipartis
- Les graphes complets
- Les arbres
- Structure de graphes

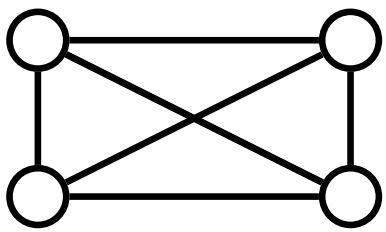
Représentation des graphes
- Matrice d'adjacences
- Liste de successeurs

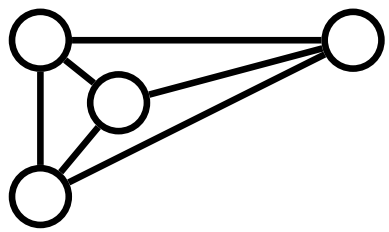
Affectation d'étudiants dans les hôpitaux
- Description & Formalisation
- Quel type d'objects il faut calculer ?
- Extension : et si les acteurs avaient des préférences

Conclusion

Les graphes planaires

Un graphe est dit planaire si tous les sommets sont adjacents deux à deux : tout couple de sommets disjoints est relié par une arête.



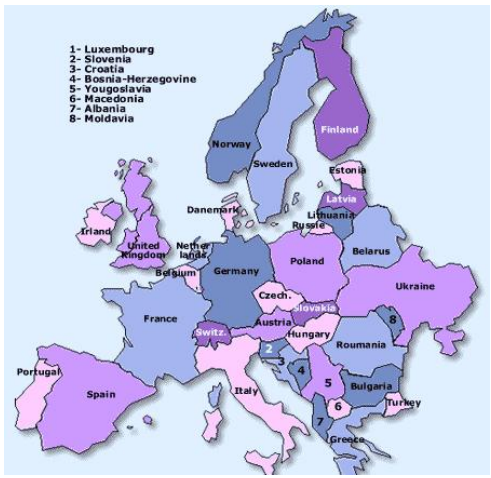


Exemple 1 : Coloriage de graphe.

- Allouer des fréquences GSM correspond à colorier les sommets d'un graphe (planaire).
  - sommets : des émetteurs radio.
  - arête entre u et v : le signal de u perturbe v ou réciproquement.
  - couleur : fréquence radio.

Exemple 1 : Coloriage de graphe.

- Allouer des fréquences GSM correspond à colorier les sommets d'un graphe (planaire).
  - sommets : des émetteurs radio.
  - arête entre u et v : le signal de u perturbe v ou réciproquement.
  - couleur : fréquence radio.
- Le problème de coloriage d'un graphe : colorier les sommets d'un graphe de telle sorte qu'il n'y ait aucune arête entre deux sommets d'une même couleur.

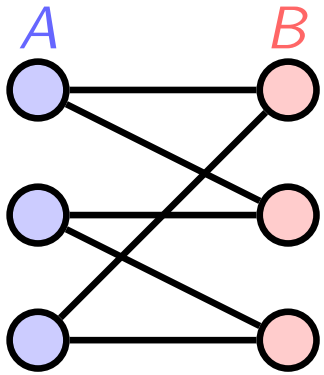


Un coloriage avec 4 couleurs

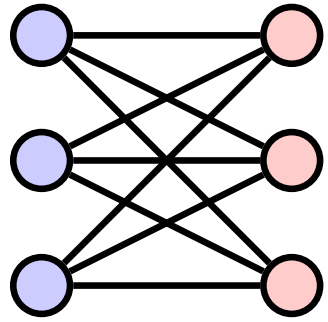
Graphes bipartis

Un graphe est biparti si son ensemble de sommets V peut être partitionné en deux sous-ensembles disjoints A et B tels que :

- Chaque arête connecte un sommet de A à un sommet de B
- Aucune arête n'existe entre les sommets à l'intérieur de A ou à l'intérieur de B



Graphe biparti complet $K_{3,3}$



Exemples du monde réel :

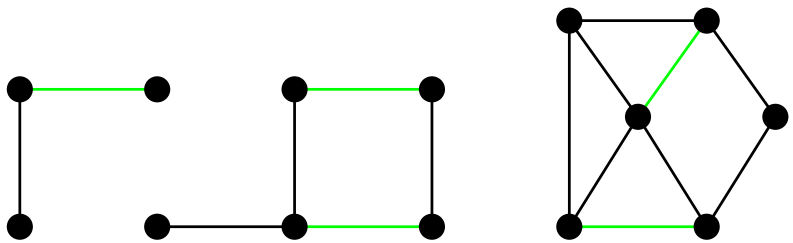
- Réseaux **Auteur-Article** dans la recherche académique
- Évaluations **Utilisateur-Film** dans les systèmes de recommandation
- Appariement **Emploi-Candidat** dans les plateformes de recrutement
- Inscriptions **Étudiant-Cours** dans les systèmes éducatifs

Exemple 2 : Les couplages de graphe.

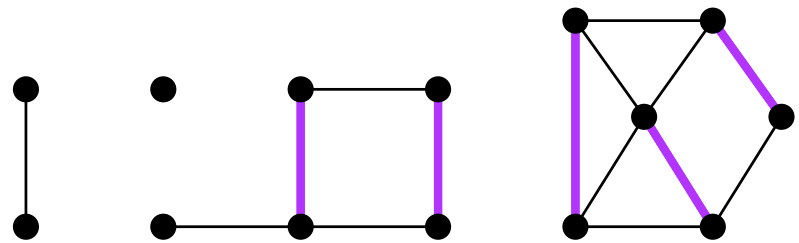
Un couplage M d'un graphe $G = (V, E)$ est un ensemble d'arêtes deux á deux non adjacentes :

$$\forall (a, a') \in M^2, \qquad a \neq a' \Rightarrow a \cap a' = \varnothing \ .$$

Un graphe peut posséder plusieurs couplages.



Un couplage maximal



Un couplage maximum

Calculer le couplage dans un graphe biparti permet d'affecter

- des tâches à des employés ;
- des étudiants à des universités (Parcours Sup (mariage stable)), etc.

Les graphes complets

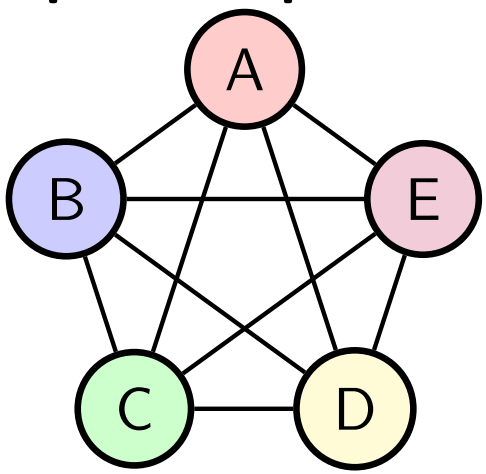
Un graphe est dit complet si tous les sommets sont adjacents deux à deux : tout couple de sommets disjoints est relié par une arête.

Les graphes complets

Un graphe est dit complet si tous les sommets sont adjacents deux à deux : tout couple de sommets disjoints est relié par une arête.

Exemple : Organisation de compétitions sportives

Dans un tournoi où chaque équipe affronte toutes les autres exactement une fois, le calendrier peut être modélisé par un **graphe complet**.



Tournoi K_5

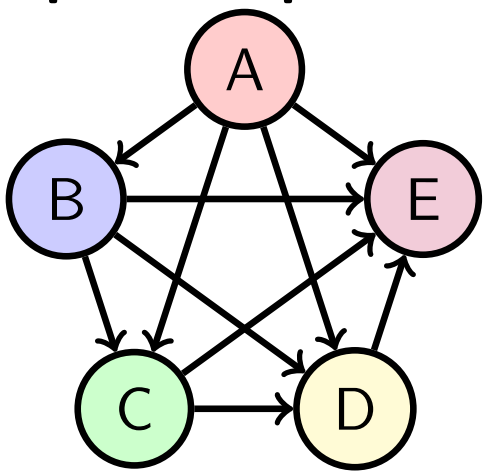
- Chaque sommet : une équipe
- Chaque arête orientée : un match
- Sens de la flèche : vainqueur
- Permet de calculer le classement final

Les graphes complets

Un graphe est dit complet si tous les sommets sont adjacents deux à deux : tout couple de sommets disjoints est relié par une arête.

Exemple : Organisation de compétitions sportives

Dans un tournoi où chaque équipe affronte toutes les autres exactement une fois, le calendrier peut être modélisé par un **graphe complet**.



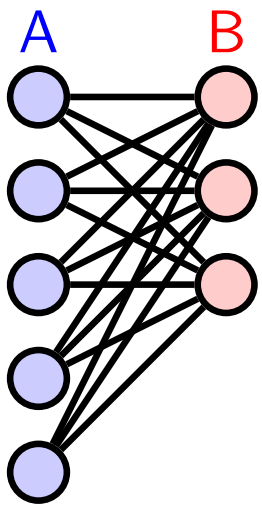
Tournoi K_5

- Chaque sommet : une équipe
- Chaque arête orientée : un match
- Sens de la flèche : vainqueur
- Permet de calculer le classement final

Les graphes bipartis complets $K_{m,n}$

Un graphe biparti $G = (A \cup B, E)$ est dit complet si :

- A et B sont deux ensembles disjoints de sommets
- Chaque sommet de A est connecté à **tous** les sommets de B

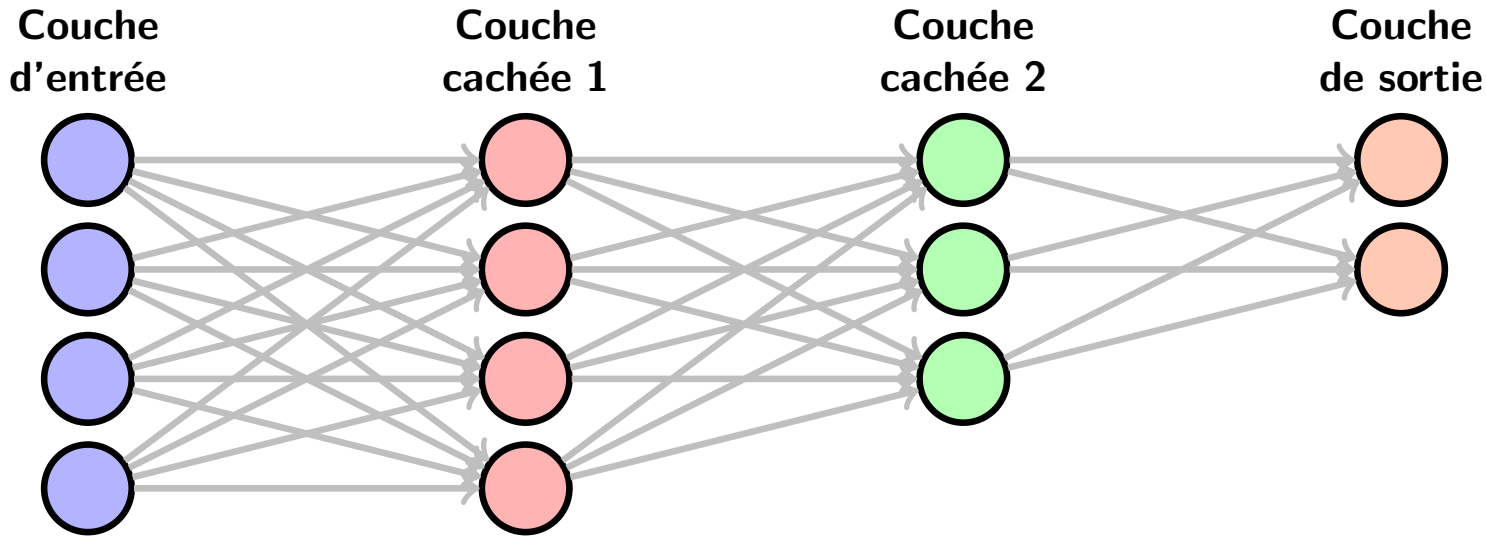


$K_{5,3}$

- Notation : $K_{m,n}$
- $m = |A|$, $n = |B|$
- Nombre d'arêtes : $m \times n$

Exemple de graphes bipartis complets

Dans les réseaux neuronaux à connexions complètes, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante, formant une structure de graphe biparti complet.

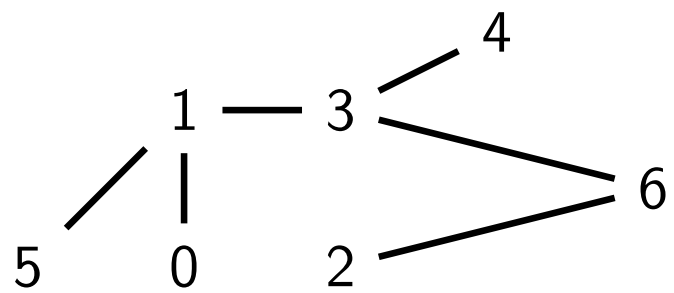


- Exemple : $K_{4,4}$ entre la couche d"entrée et la couche cachée 1,
- Remarque : les poids des arêtes correspondent aux paramètres d'apprentissage du réseau.

Les arbres sont partout !

- Un graphe connexe sans cycle est appelé un arbre.
- Un graphe sans-cycle est appelé une forêt :
  - chacune de ses composantes connexes est un arbre.

- Dès qu'on a des objets, des relations entre objets, et pas de cycle, on a donc un arbre ou une forêt.

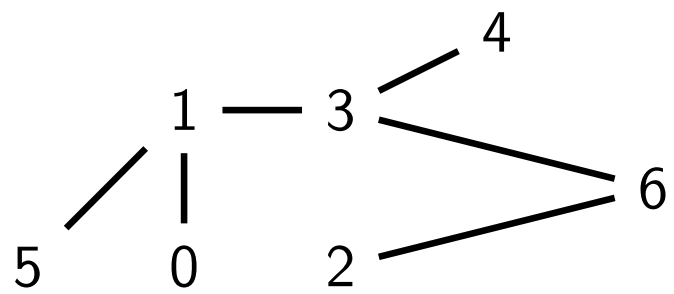


- Les arbres sont omniprésents en informatique.

Les arbres sont partout !

- Un graphe connexe sans cycle est appelé un arbre.
- Un graphe sans-cycle est appelé une forêt :
  - chacune de ses composantes connexes est un arbre.

- Dès qu'on a des objets, des relations entre objets, et pas de cycle, on a donc un arbre ou une forêt.

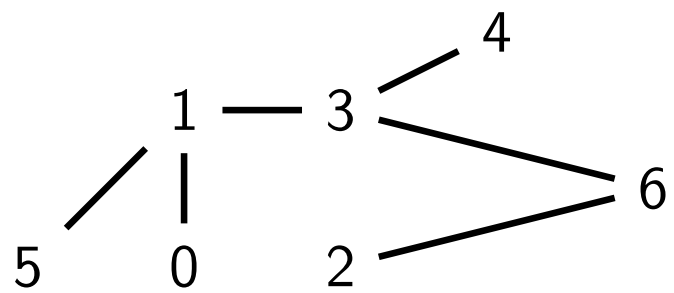


- Les arbres sont omniprésents en informatique.

Les arbres sont partout !

- Un graphe connexe sans cycle est appelé un arbre.
- Un graphe sans-cycle est appelé une forêt :
  - chacune de ses composantes connexes est un arbre.

- Dès qu'on a des objets, des relations entre objets, et pas de cycle, on a donc un arbre ou une forêt.

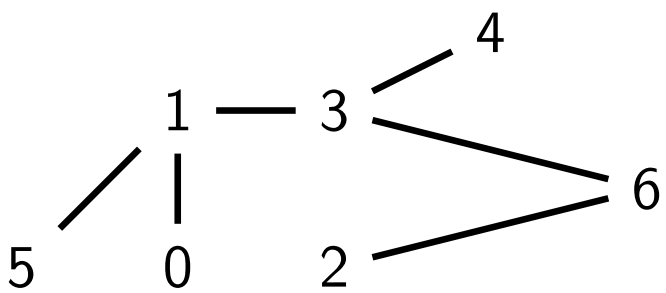


- Les arbres sont omniprésents en informatique.

Quelques propriétés sur les arbres

Soit $G = (V, E)$ un graphe.
Les propriétés suivantes sont équivalentes :

- G est un arbre ;
- G est connexe, mais ne l'est plus si on enlève n'importe laquelle de ses arêtes ;
- G est connexe et $|E| = |V| - 1$;
- G est sans cycle et $|E| = |V| - 1$;



Propriété

Propriété

Si G est un arbre, alors G est connexe et $|E| = |V| - 1$

Preuve : par récurrence sur le nombre d'arêtes m.

- si $m = 0$: c'est vrai (le graphe = un sommet)
- Supposons que ce soit vrai pour les arbres de m arêtes ou moins.
  1. Soit T un arbre avec $m + 1$ arêtes.
  2. Soit a une arête de T : Construisons A l'arbre T privé de a
  3. A a deux composantes connexes A_1 et A_2.

Propriété

Propriété

Si G est un arbre, alors G est connexe et $|E| = |V| - 1$

Preuve : par récurrence sur le nombre d'arêtes m.

- si $m = 0$: c'est vrai (le graphe = un sommet)
- Supposons que ce soit vrai pour les arbres de m arêtes ou moins.
  1. Soit T un arbre avec $m + 1$ arêtes.
  2. Soit a une arête de T : Construisons A l'arbre T privé de a
  3. A a deux composantes connexes A_1 et A_2.

Propriété

Propriété

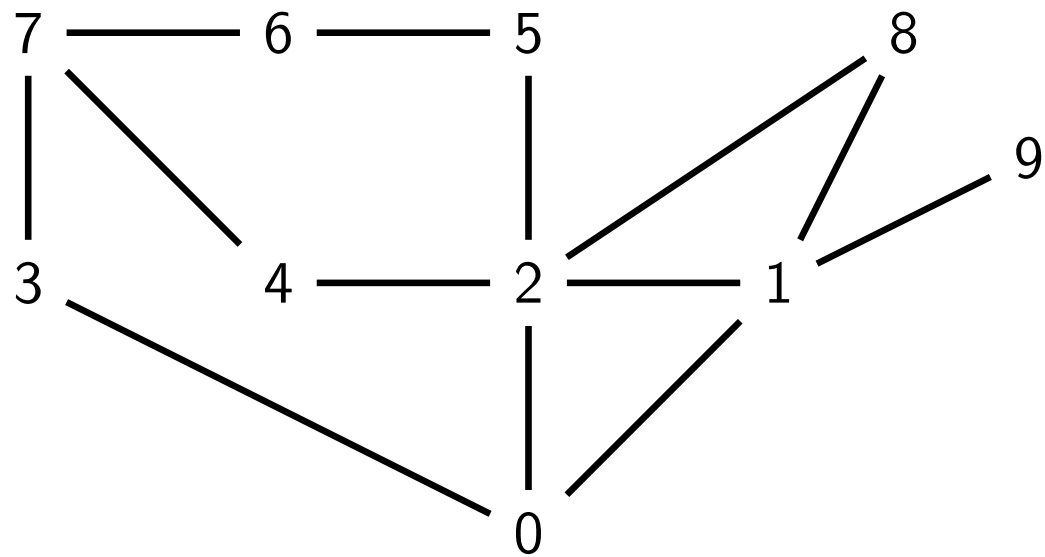
Si G est un arbre, alors G est connexe et $|E| = |V| - 1$

Preuve : par récurrence sur le nombre d'arêtes m.

- si $m = 0$: c'est vrai (le graphe = un sommet)
- Supposons que ce soit vrai pour les arbres de m arêtes ou moins.
  1. Soit T un arbre avec $m + 1$ arêtes.
  2. Soit a une arête de T : Construisons A l'arbre T privé de a
  3. A a deux composantes connexes A_1 et A_2.
  4. On applique l'hypothèse de récurrence à A_1 et à A_2 car $m_1 = n_1 - 1$ et $m_2 = n_2 - 1$
  5. $m_1 + m_2 + 1 = m$ $n_1 + n_2 = n$
  6. Donc $m = n_1 - 1 + n_2 - 1 + 1 = n - 1$

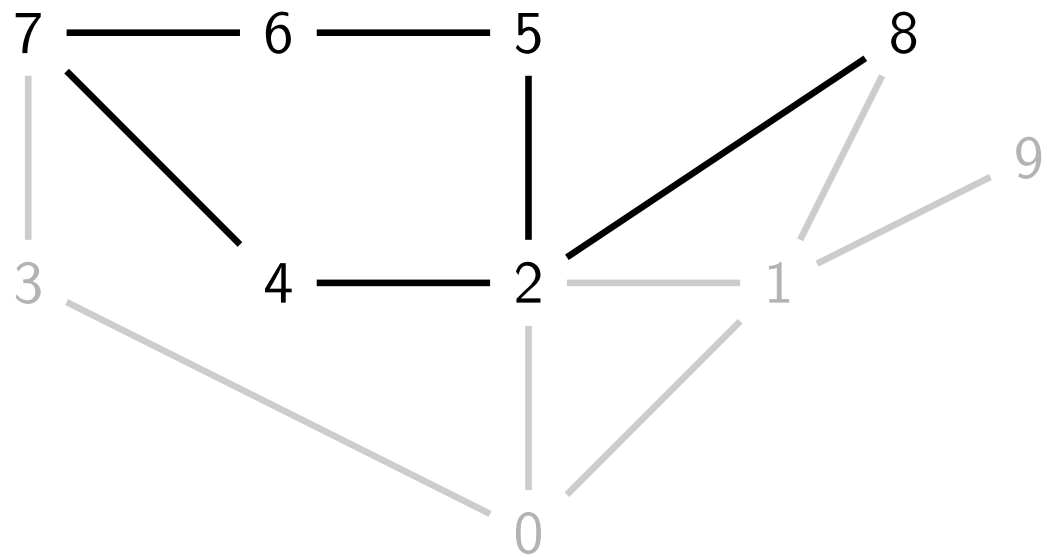
Structure de graphes : sous-graphe

Un sous-graphe de G consiste à considérer seulement une partie des sommets de V et les arêtes induites par E.



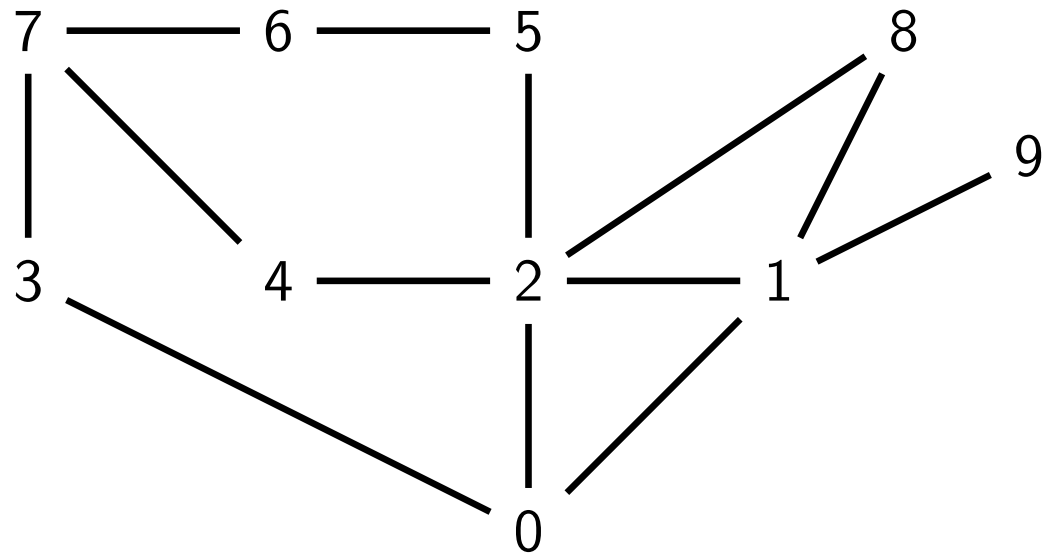
Structure de graphes : sous-graphe

Un sous-graphe de G consiste à considérer seulement une partie des sommets de V et les arêtes induites par E.



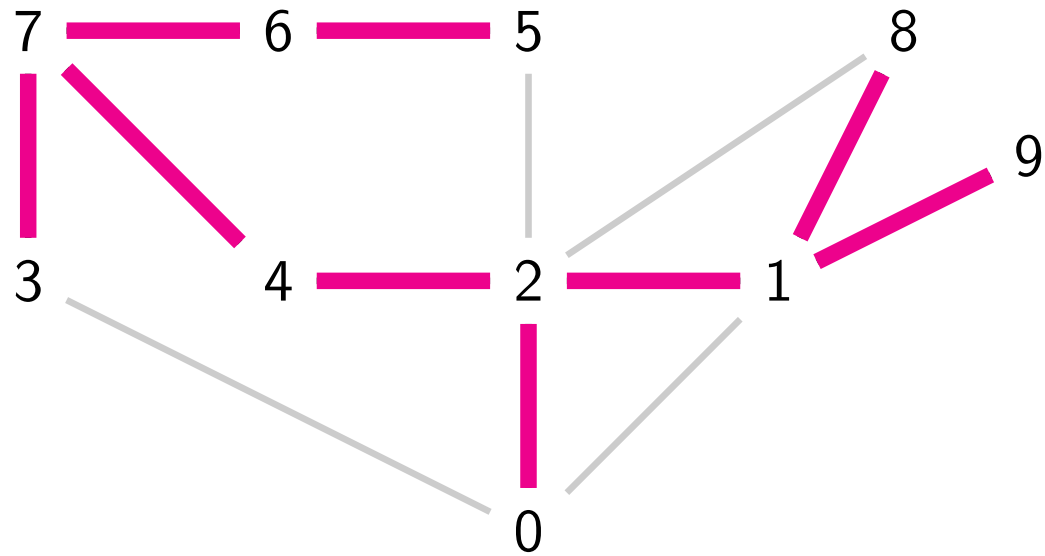
Structure de graphes : graphe partiel

Un graphe partiel de G consiste à ne considérer qu'une partie des arêtes de E.



Structure de graphes : graphe partiel

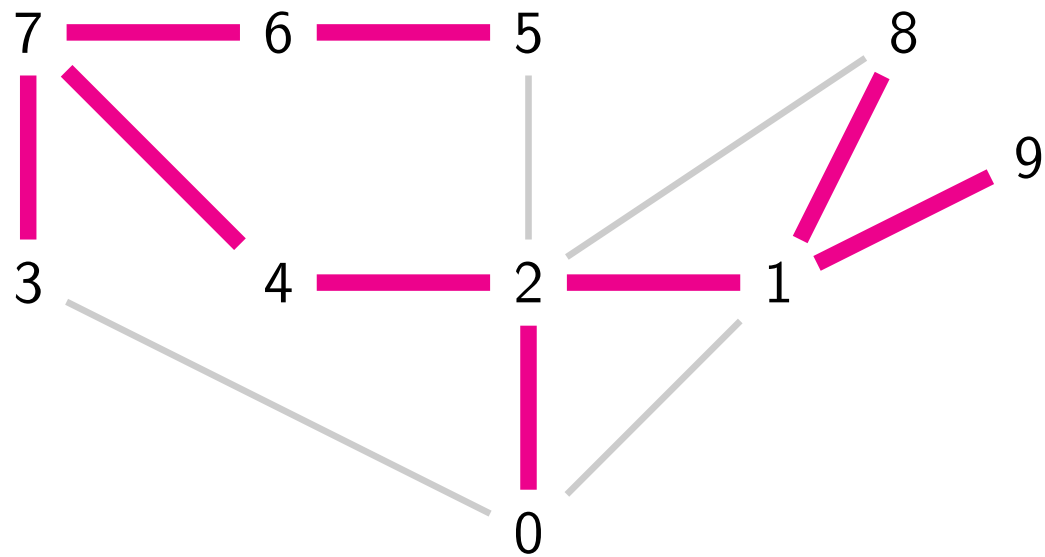
Un graphe partiel de G consiste à ne considérer qu'une partie des arêtes de E.



Remarques

Soit $G = (V, E)$ avec $V = \{1, 2, \ldots, n\}$,

- G est connexe si il a possède comme un graphe partiel un arbre couvrant tous les sommets



Plan

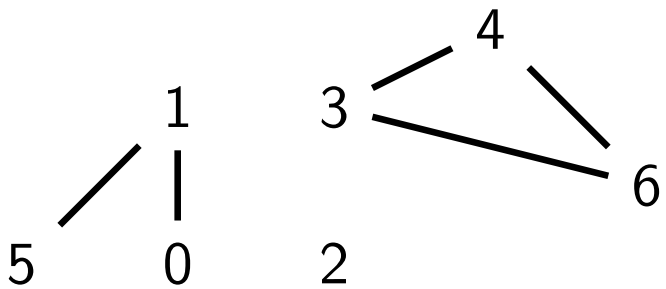
- Les graphes : un outil pour la modélisation
- Terminologie sur les graphes
- Les graphes classiques
  - Les graphes planaires
  - Graphes bipartis
  - Les graphes complets
  - Les arbres
  - Structure de graphes
- Représentation des graphes
  - Matrice d'adjacences
  - Liste de successeurs
- Affectation d'étudiants dans les hôpitaux
  - Description & Formalisation
  - Quel type d'objects il faut calculer ?
  - Extension : et si les acteurs avaient des préférences
- Conclusion

Représentation des graphes : matrice d'adjacence

Soit $G = (V, E)$ avec $V = \{1, 2, \ldots, n\}$,

- G peut être représenté par une matrice M $n \times n$.

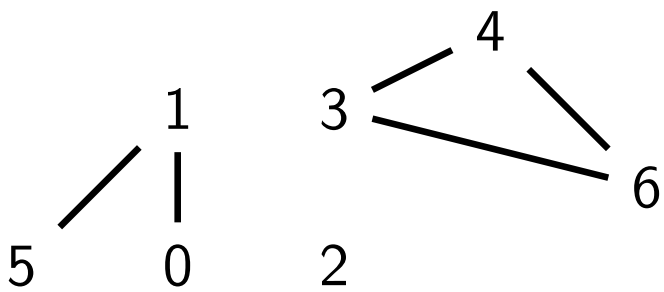
$$M_{i,j} = \begin{cases} 1 & \text{si } (i,j) \in E \\ 0 & \text{sinon} \end{cases}$$



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

Représentation des graphes : liste de successeurs

- On associe à chaque sommet i, la liste des sommets j tels que $(i, j) \in E$.



- $L[0] = (1)$
- $L[1] = (0, 5)$
- $L[2] = ()$
- $L[3] = (4, 6)$
- $L[4] = (3, 6)$
- $L[5] = (1)$
- $L[6] = (3, 4)$

Meilleure représentation ?

- Matrice : mémoire $O(n^2)$
- Listes : mémoire $O(n+m)$

  où n nombre de sommets, m nombre d'arêtes.

- Quelle est la meilleure représentation ?

  Cela dépend du contexte.

	La méthode plus efficace est
Tester si (u, v) est dans le graphe.	matrice d'adjacences
Tester si calculer le degré de v	liste de successeurs
Stocker des graphes denses	matrice d'adjacences
Stocker des graphes creux	liste de successeurs
Insérer ou supprimer des arêtes	matrice d'adjacences

Les réseaux du monde réel sont des graphes creux.

Observation clé : La plupart des réseaux du monde réel sont **creux (sparses)** :

$$\text{Degré moyen} \ll |V| - 1$$

Réseau	**Nœuds**	**Arêtes**	**Type**	**Degré moyen**	**Densité**
Internet (niveau AS)	50 000+	200 000+	Non orienté	∼8	$3,2 \times 10^{-4}$
Réseau de citations	3M+ articles	30M+ citations	Orienté	∼10	$3,3 \times 10^{-6}$
Réseau social	2,9G utilisateurs	500G+ connexions	Orienté	∼170	$5,9 \times 10^{-8}$
Interactions protéiques	20K+ protéines	300K+ interactions	Non orienté	∼15	$1,5 \times 10^{-3}$

- Densité $= \frac{2|E|}{|V|(|V|-1)}$ (non orienté) ou $\frac{|E|}{|V|(|V|-1)}$ (orienté)
- Même les réseaux massifs ont un nombre limité de connexions par nœud
- Permet un stockage efficace (listes d'adjacence)

Plan

- Les graphes : un outil pour la modélisation
- Terminologie sur les graphes
- Les graphes classiques
  - Les graphes planaires
  - Graphes bipartis
  - Les graphes complets
  - Les arbres
  - Structure de graphes
- Représentation des graphes
  - Matrice d'adjacences
  - Liste de successeurs
- Affectation d'étudiants dans les hôpitaux
  - Description & Formalisation
  - Quel type d'objects il faut calculer ?
  - Extension : et si les acteurs avaient des préférences
- Conclusion

Affectation d'étudiants dans les hôpitaux

- un ensemble $\mathcal{H}$ de hôpitaux

  Chaque hôpital **u** a un poste à pouvoir.

- un ensemble $\mathcal{E}$ d'étudiants.

  Chaque étudiant **s** postule à un ensemble $\mathcal{X}_{\mathbf{s}}$ de postes.

Objectif : trouver une affectation d'étudiants respectant certaines contraintes.

Par exemple :

- $\mathcal{H} = \{Avignon, Bordeaux, Cachan\}$
- $\mathcal{E} = \{Sandrine, Teo, Zoe\}$
- $\mathcal{X}_{Sandrine} = \{Avignon, Bordeaux\}$, $\mathcal{X}_{Teo} = \{Bordeaux, Cachan\}$, $\mathcal{X}_{Zoe} = \{Avignon, Bordeaux\}$.

Affectation d'étudiants dans les hôpitaux

- un ensemble $\mathcal{H}$ de hôpitaux

  Chaque hôpital **u** a un poste à pouvoir.

- un ensemble $\mathcal{E}$ d'étudiants.

  Chaque étudiant **s** postule à un ensemble $\mathcal{X}_{\mathbf{s}}$ de postes.

Objectif : trouver une affectation d'étudiants respectant certaines contraintes.

Par exemple :

- $\mathcal{H} = \{Avignon, Bordeaux, Cachan\}$
- $\mathcal{E} = \{Sandrine, Teo, Zoe\}$
- $\mathcal{X}_{Sandrine} = \{Avignon, Bordeaux\}$, $\mathcal{X}_{Teo} = \{Bordeaux, Cachan\}$, $\mathcal{X}_{Zoe} = \{Avignon, Bordeaux\}$.

Affectation d'étudiants dans les hôpitaux

En entrée :

- un ensemble $\mathcal{H}$ de hôpitaux
- un ensemble $\mathcal{E}$ d'étudiants.

Objectif : trouver une affectation d'étudiants

Question : Comment modéliser ce problème ?

Sous forme de graphe $G = (V, E)$

- $V = \mathcal{E} \cup \mathcal{H}$
- $e = (\mathbf{s}, \mathbf{u}) \in E$ si $\mathbf{s}$ postule chez $\mathbf{u}$

Question : Comment modéliser une affectation ?

Affectation d'étudiants dans les hôpitaux

En entrée :

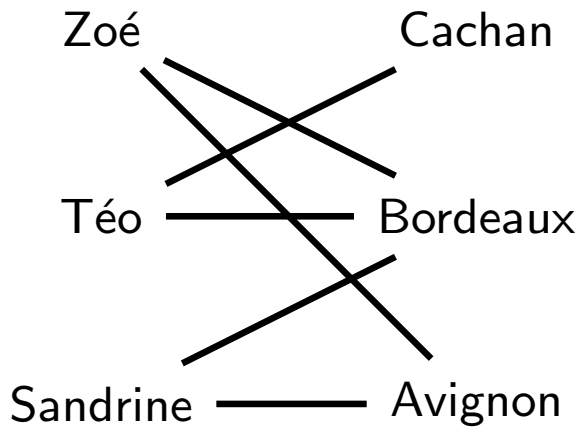
- un ensemble $\mathcal{H}$ de hôpitaux
- un ensemble $\mathcal{E}$ d'étudiants.

Objectif : trouver une affectation d'étudiants

Question : Comment modéliser ce problème ?

Sous forme de graphe $G = (V, E)$

- $V = \mathcal{E} \cup \mathcal{H}$
- $e = (\mathbf{s}, \mathbf{u}) \in E$ si $\mathbf{s}$ postule chez $\mathbf{u}$



Question : Comment modéliser une affectation ?

Affectation d'étudiants dans les hôpitaux

En entrée :

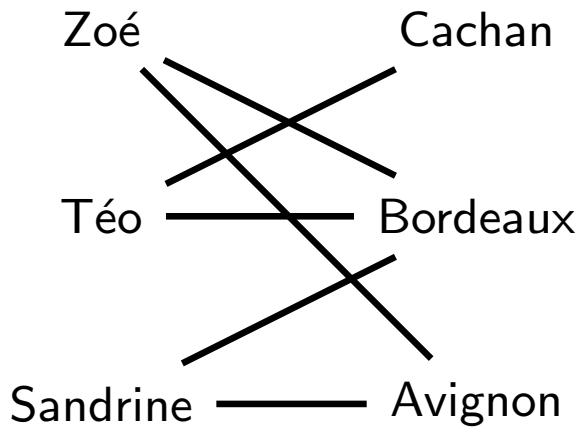
- un ensemble $\mathcal{H}$ de hôpitaux
- un ensemble $\mathcal{E}$ d'étudiants.

Objectif : trouver une affectation d'étudiants

Question : Comment modéliser ce problème ?

Sous forme de graphe $G = (V, E)$

- $V = \mathcal{E} \cup \mathcal{H}$
- $e = (\mathbf{s}, \mathbf{u}) \in E$ si $\mathbf{s}$ postule chez $\mathbf{u}$



Question : Comment modéliser une affectation ?

Définition d'un couplage

Soit un graphe simple non orienté $G = (V, E)$
Un couplage M est un ensemble d'arêtes deux à deux non adjacentes :

$$\forall (a, a') \in M^2, \qquad a \neq a' \Rightarrow a \cap a' = \varnothing \ .$$

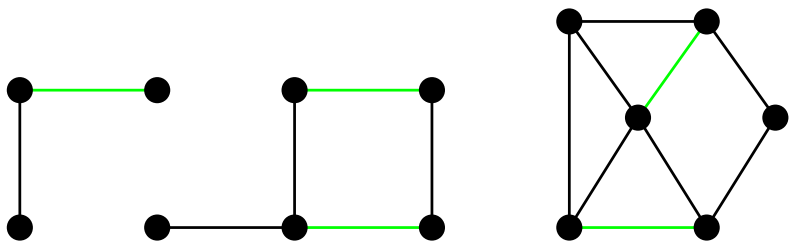
Terminologies :

- Les sommets incidents à une arête de G sont les sommets couplés, apparié ou couverts par M.
- Le couplage M sature U, si U est un ensemble des sommets couplés par M.

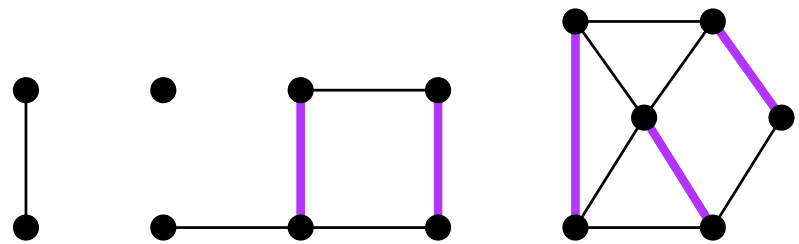
Les différents couplages

- Un couplage maximal est un couplage M ayant la propriété que si une arête e est ajoutée, alors $M \cup \{e\}$ n'en est pas.
- Un couplage maximum est un couplage contenant le plus grand nombre possible d'arêtes.

Un graphe peut posséder plusieurs couplages maximum.



Couplage maximal



Couplage maximum

Affectation d'étudiants dans les hôpitaux

En entrée :

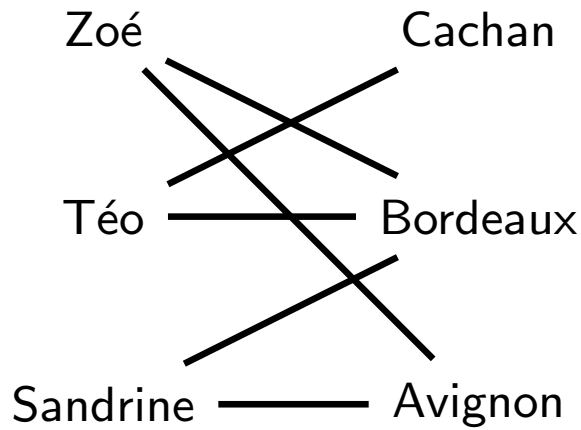
- un ensemble $\mathcal{H}$ de hôpitaux
- un ensemble $\mathcal{E}$ d'étudiants.

Objectif : trouver une affectation d'étudiants

Question : Comment modéliser ce problème ?

Sous forme de graphe $G = (V, E)$

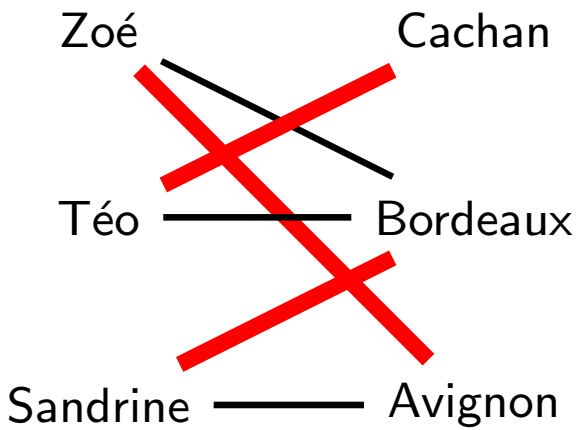
- $V = \mathcal{E} \cup \mathcal{H}$
- $e = (\mathbf{s}, \mathbf{u}) \in E$ si $\mathbf{s}$ postule chez $\mathbf{u}$

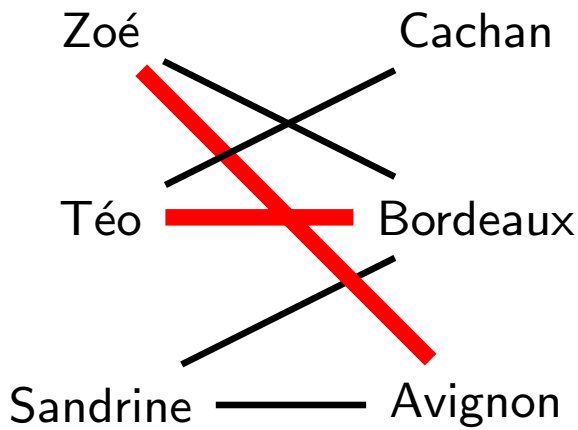


Question : Comment modéliser une affectation ? un couplage

Remarques

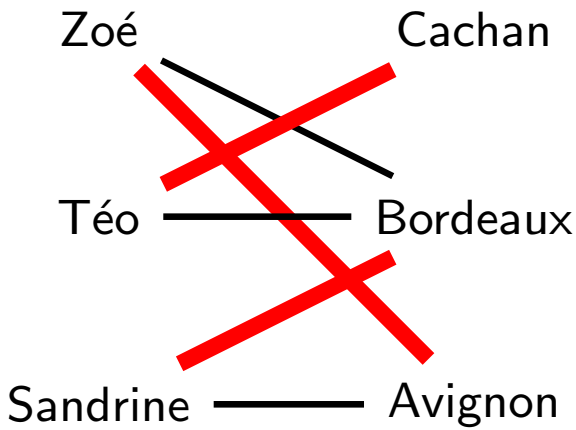
Observation : Il existe plusieurs affectations.

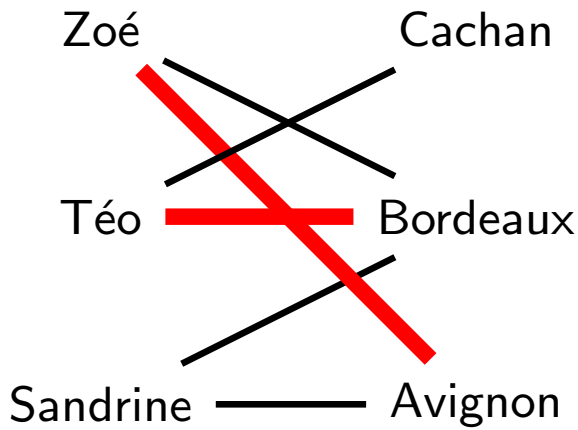




Remarques

Observation : Il existe plusieurs affectations.



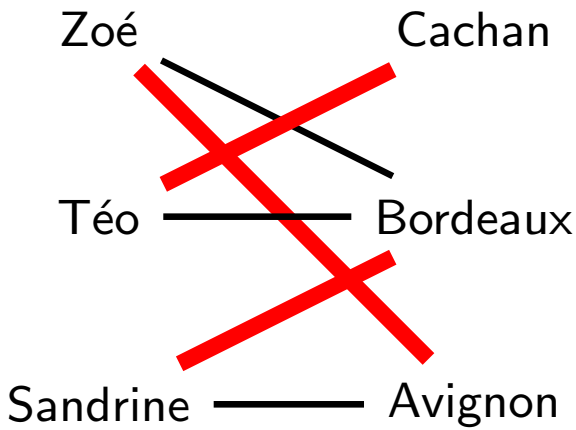


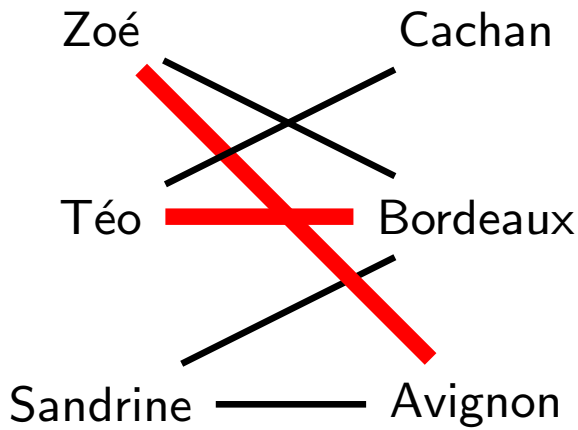
Question : Comment choisir une affectation ?

en considérant une fonction à optimiser.

Remarques

Observation : Il existe plusieurs affectations.



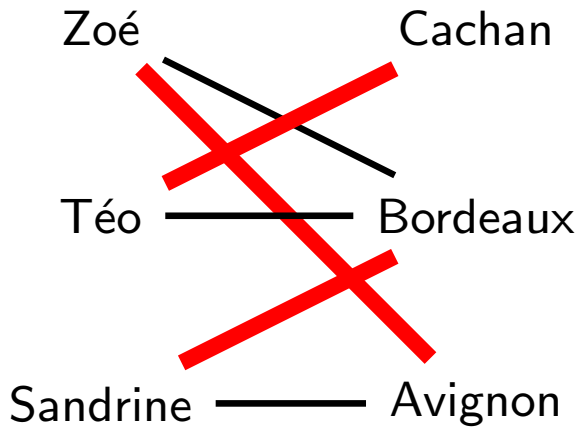


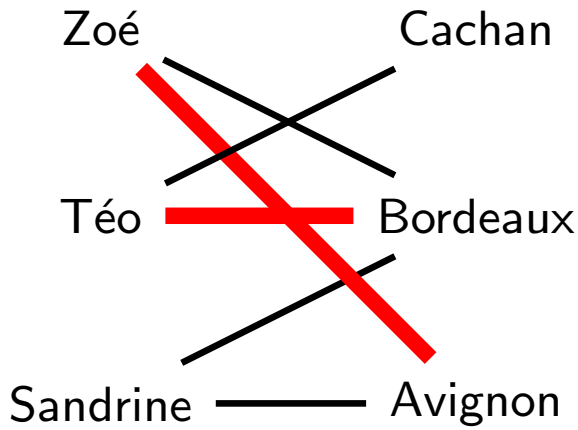
Question : Comment choisir une affectation ?

en maximisant le nombre de postes à pourvoir.

Remarques

Observation : Il existe plusieurs affectations.





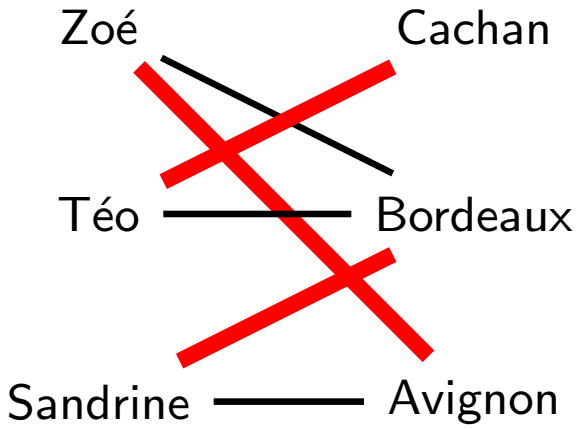
Question : Comment choisir une affectation ?

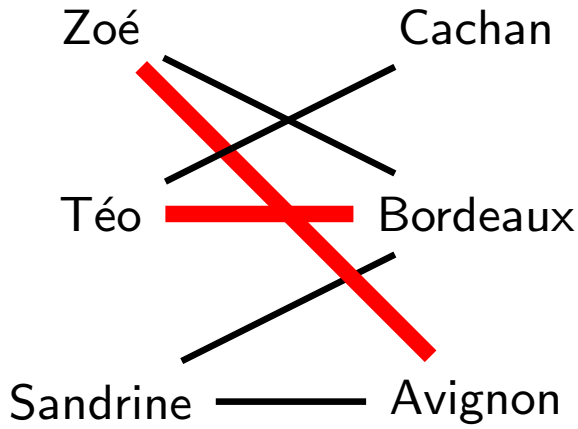
en maximisant le nombre de postes à pourvoir.

Donc il faut construire un couplage maximum.

Remarques

Observation : Il existe plusieurs affectations.





Question : Comment choisir une affectation ?

en maximisant le nombre de postes à pourvoir.

Donc il faut construire un couplage maximum.

En utilisant la méthode hondroise en $\mathcal{O}(|V|^3)$ opérations

Affectation d'étudiants dans les hôpitaux

En entrée :

- un ensemble $\mathcal{H}$ de hôpitaux
- un ensemble $\mathcal{E}$ d'étudiants.

Objectif : trouver une affectation d'étudiants

Affectation d'étudiants dans les hôpitaux

En entrée :

- un ensemble $\mathcal{H}$ de hôpitaux
- un ensemble $\mathcal{E}$ d'étudiants.
- Chaque acteur (hôpital/étudiant) v a des préférences

l'acteur v préfère a à b $\quad a \succ_v b$

Objectif : trouver une affectation d'étudiants

en tenant compte des préférences des acteurs

Affectation d'étudiants dans les hôpitaux

En entrée :

- un ensemble $\mathcal{H}$ de hôpitaux
- un ensemble $\mathcal{E}$ d'étudiants.
- Chaque acteur (hôpital/étudiant) v a des préférences

l'acteur v préfère a à b $\quad a \succ_v b$

Objectif : trouver une affectation d'étudiants

en tenant compte des préférences des acteurs

Question : Comment tenir compte des préférences ?

Comment tenir compte des préférences ?

$A \succ B \succ C$	Zoé	Cachan	$S \succ T \succ Z$
$A \succ B \succ C$	Téo	Bordeaux	$T \succ S \succ Z$
$B \succ A \succ C$	Sandrine	Avignon	$S \succ T \succ Z$

Plusieurs solutions existent : Trouver une affectation d'étudiants

- en prenant les k premiers choix des universités.

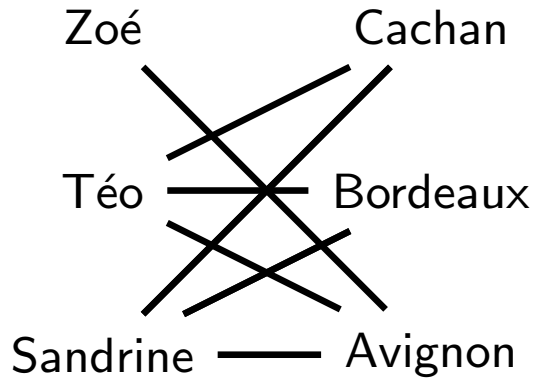
Comment tenir compte des préférences ?

$A \succ B \succ C$	Zoé	Cachan	$S \succ T \succ Z$
$A \succ B \succ C$	Téo	Bordeaux	$T \succ S \succ Z$
$B \succ A \succ C$	Sandrine	Avignon	$S \succ T \succ Z$

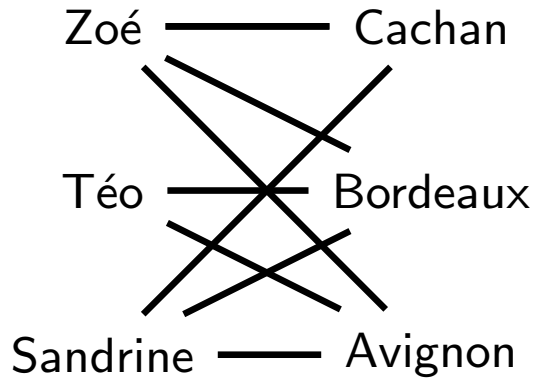
Plusieurs solutions existent : Trouver une affectation d'étudiants

- en prenant les k premiers choix des universités.

Pour $k = 1$



Pour $k = 2$:



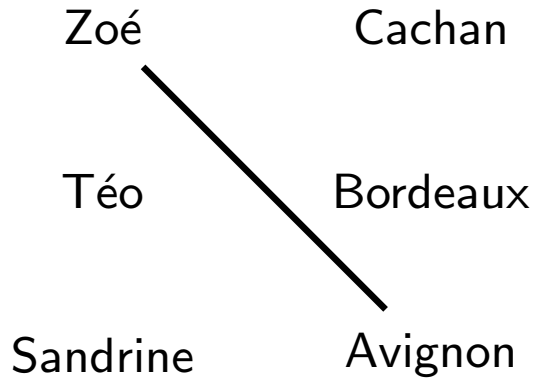
Comment tenir compte des préférences ?

$A \succ B \succ C$	Zoé	Cachan	$S \succ T \succ Z$
$A \succ B \succ C$	Téo	Bordeaux	$T \succ S \succ Z$
$B \succ A \succ C$	Sandrine	Avignon	$S \succ T \succ Z$

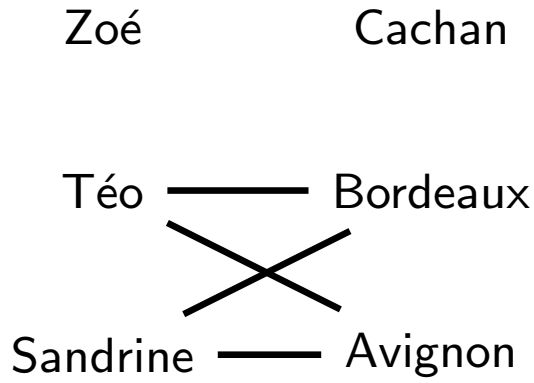
Plusieurs solutions existent : Trouver une affectation d'étudiants

- en prenant les k premiers choix des universités.

Pour $k = 1$



Pour $k = 2$:



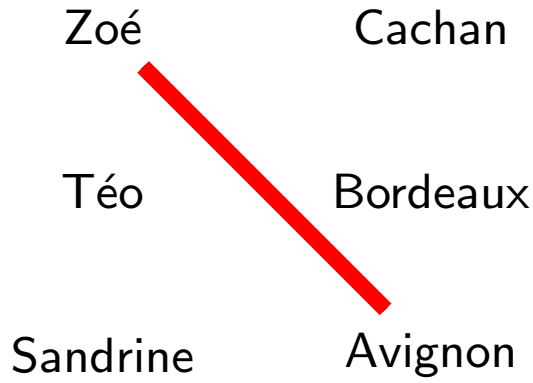
Comment tenir compte des préférences ?

$A \succ B \succ C$	Zoé	Cachan	$S \succ T \succ Z$
$A \succ B \succ C$	Téo	Bordeaux	$T \succ S \succ Z$
$B \succ A \succ C$	Sandrine	Avignon	$S \succ T \succ Z$

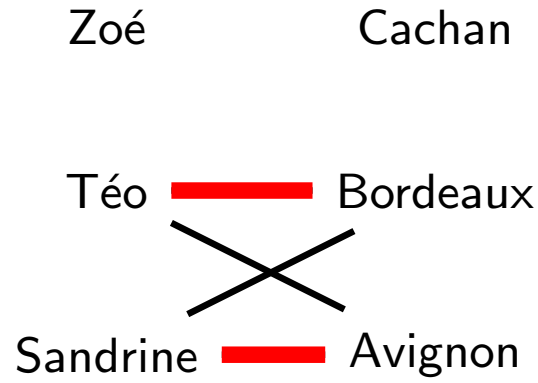
Plusieurs solutions existent : Trouver une affectation d'étudiants

- en prenant les k premiers choix des universités.

Pour $k = 1$



Pour $k = 2$:



Comment tenir compte des préférences ?

$A \succ B \succ C$	Zoé	Cachan	$S \succ T \succ Z$
$A \succ B \succ C$	Téo	Bordeaux	$T \succ S \succ Z$
$B \succ A \succ C$	Sandrine	Avignon	$S \succ T \succ Z$

Plusieurs solutions existent : Trouver une affectation d'étudiants

- en prenant les k premiers choix des universités.

Comment tenir compte des préférences ?

$A \succ B \succ C$	Zoé	Cachan	$S \succ T \succ Z$
$A \succ B \succ C$	Téo	Bordeaux	$T \succ S \succ Z$
$B \succ A \succ C$	Sandrine	Avignon	$S \succ T \succ Z$

Plusieurs solutions existent : Trouver une affectation d'étudiants

- en prenant les k premiers choix des universités.
  - Construire le graphe $G = (V, E)$ avec $V = \mathcal{E} \cup \mathcal{H}$ et si $e = (\mathbf{s}, \mathbf{u}) \in E$ si $\mathbf{s}$ postule chez $\mathbf{u}$

Comment tenir compte des préférences ?

$A \succ B \succ C$	Zoé	Cachan	$S \succ T \succ Z$
$A \succ B \succ C$	Téo	Bordeaux	$T \succ S \succ Z$
$B \succ A \succ C$	Sandrine	Avignon	$S \succ T \succ Z$

Plusieurs solutions existent : Trouver une affectation d'étudiants

- en prenant les k premiers choix des universités.
  - Construire le graphe $G = (V, E)$ avec $V = \mathcal{E} \cup \mathcal{H}$ et si $e = (\mathbf{s}, \mathbf{u}) \in E$ si $\mathbf{s}$ postule chez $\mathbf{u}$ et $\mathbf{s}$ est dans le k premiers choix de l'université.

Comment tenir compte des préférences ?

$A \succ B \succ C$	Zoé	Cachan	$S \succ T \succ Z$
$A \succ B \succ C$	Téo	Bordeaux	$T \succ S \succ Z$
$B \succ A \succ C$	Sandrine	Avignon	$S \succ T \succ Z$

Plusieurs solutions existent : Trouver une affectation d'étudiants

- en prenant les k premiers choix des universités.
  - Construire le graphe $G = (V, E)$ avec $V = \mathcal{E} \cup \mathcal{H}$ et si $e = (\mathbf{s}, \mathbf{u}) \in E$ si $\mathbf{s}$ postule chez $\mathbf{u}$ et $\mathbf{s}$ est dans le k premiers choix de l'université.
  - Construire un couplage maximum

Comment tenir compte des préférences ?

$A \succ B \succ C$	Zoé	Cachan	$S \succ T \succ Z$
$A \succ B \succ C$	Téo	Bordeaux	$T \succ S \succ Z$
$B \succ A \succ C$	Sandrine	Avignon	$S \succ T \succ Z$

Plusieurs solutions existent : Trouver une affectation d'étudiants

- en prenant les k premiers choix des universités.
- en prenant les k premiers choix de tous les acteurs.
- ....

Comment tenir compte des préférences ?

$A \succ B \succ C$	Zoé	Cachan	$S \succ T \succ Z$
$A \succ B \succ C$	Téo	Bordeaux	$T \succ S \succ Z$
$B \succ A \succ C$	Sandrine	Avignon	$S \succ T \succ Z$

Plusieurs solutions existent : Trouver une affectation d'étudiants

- en prenant les k premiers choix des universités.
- en prenant les k premiers choix de tous les acteurs.
- ....

Question : et si les étudiants ne sont pas coopératifs ?
Reformulation : Compte tenu d'un ensemble de préférences parmi les hôpitaux et les étudiants, concevoir un processus d'admission décentralisé.

Affectation d'étudiants dans les hôpitaux

Objectif : Compte tenu d'un ensemble de préférences parmi les hôpitaux et les étudiants, concevoir un processus d'admission décentralisé.

Un couple $(\mathbf{s}, \mathbf{u})$ est une paire bloquante pour un couplage $\mathcal{M}$ si :

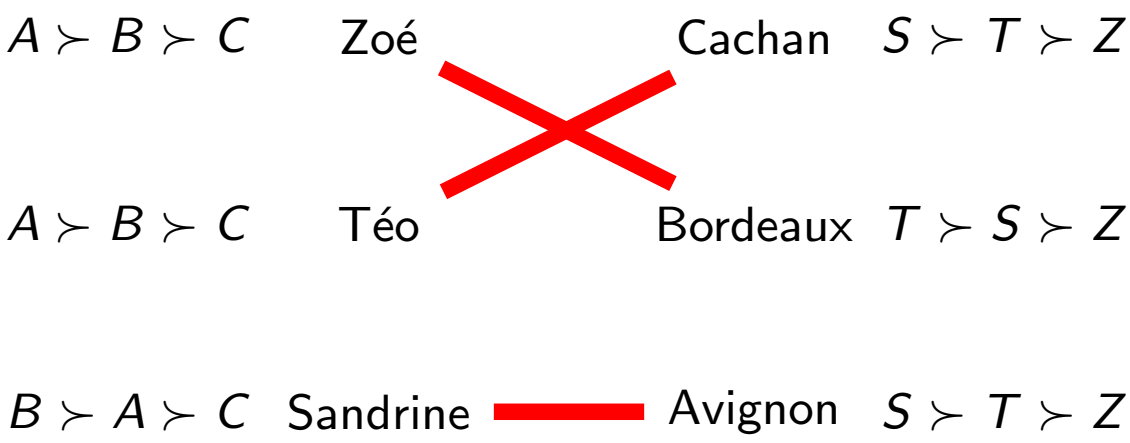
1. $(\mathbf{s}, \mathbf{u}) \notin \mathcal{M}$,
2. $\mathbf{s} \succ_{\mathbf{u}} \mathcal{M}(\mathbf{u})$
3. $\mathbf{u} \succ_{\mathbf{s}} \mathcal{M}(\mathbf{s})$

Affectation stable : Affectation sans aucune paire bloquante.

- Condition naturelle et souhaitable.
- L'intérêt personnel individuel empêche tout accord entre l'hôpital et l'étudiant.

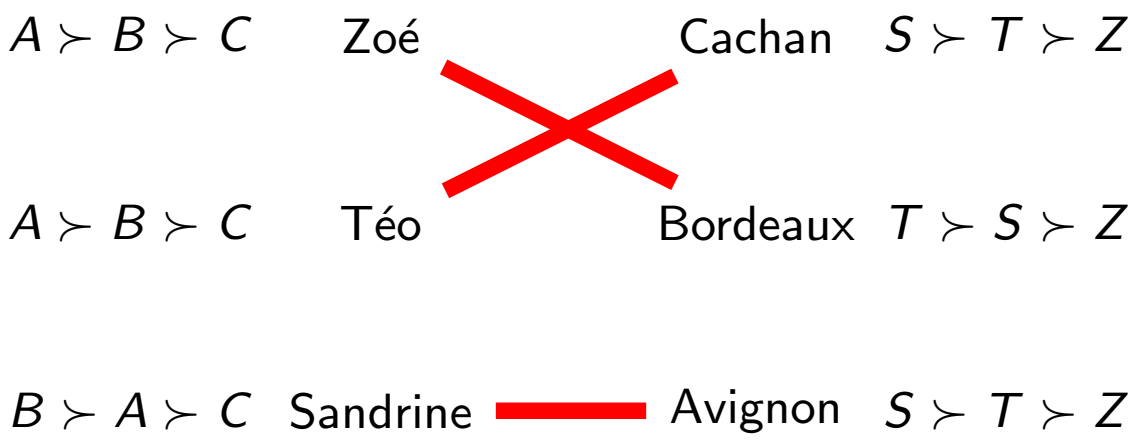
Exemple : Affectation d'étudiants

- $\mathcal{H} = \{Avignon, Bordeaux, Cachan\}$
- $\mathcal{E} = \{Sandrine, Teo, Zoe\}$



Exemple : Affectation d'étudiants

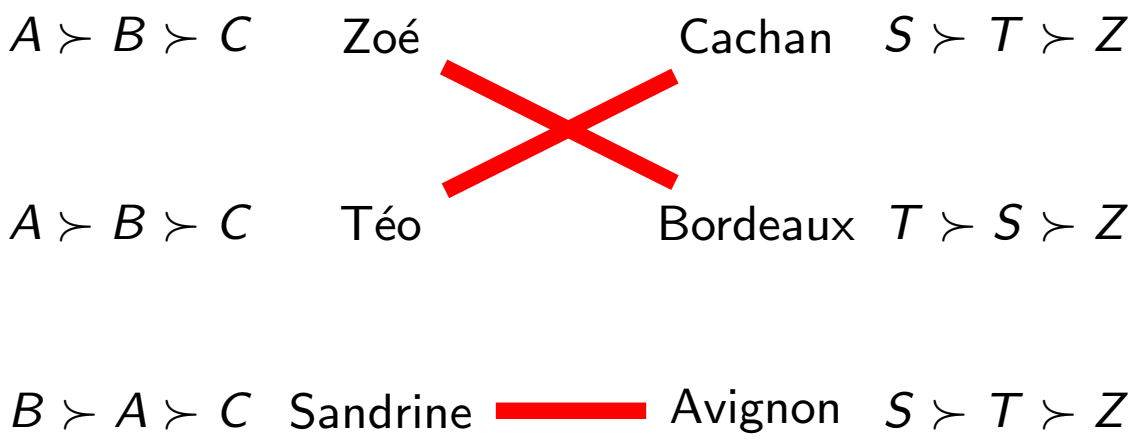
- $\mathcal{H} = \{Avignon, Bordeaux, Cachan\}$
- $\mathcal{E} = \{Sandrine, Teo, Zoe\}$



Les paires suivantes $A - T$, $B - S$, $C - Z$ sont-elles bloquantes ?

Exemple : Affectation d'étudiants

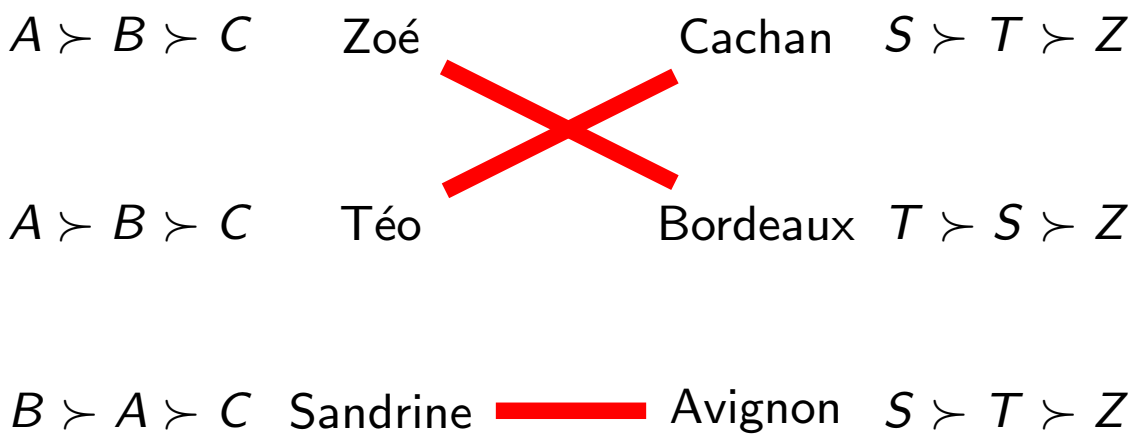
- $\mathcal{H} = \{Avignon, Bordeaux, Cachan\}$
- $\mathcal{E} = \{Sandrine, Teo, Zoe\}$



Les paires suivantes $A - T$, $B - S$, $C - Z$ sont-elles bloquantes ? non

Exemple : Affectation d'étudiants

- $\mathcal{H} = \{Avignon, Bordeaux, Cachan\}$
- $\mathcal{E} = \{Sandrine, Teo, Zoe\}$



Les paires suivantes $A - T$, $B - S$, $C - Z$ sont-elles bloquantes ? non
C'est une affectation stable

Algorithme de Gale-Shapley

1. initialise le couplage $\mathcal{M}$ comme le couplage vide
2. tant qu'il existe un hôpital **u** non-affecté et qu'il n'a pas demandé à tout sa liste faire
   - 2.1 **s** $\leftarrow$ le premier étudiant de sa liste que **u** n'a pas encore demandé
   - 2.2 si **s** n'est pas saturé alors
     insérer le couple $(\mathbf{s}, \mathbf{u})$ dans le couplage
   - 2.3 Sinon
     - 2.3.1 si **s** préfère **u** à sa demande courante $\mathbf{u}' = \mathcal{M}(\mathbf{s})$
       alors remplacer le couple $(\mathbf{s}, \mathbf{u}')$ par $(\mathbf{s}, \mathbf{u})$
       sinon rejeter cette demande
3. Retourner le couplage $\mathcal{M}$

Exemple : Exécution de l'algorithme de Gale-Shapley

$A \succ B \succ C$	Zoé	Cachan	$S \succ T \succ Z$
$A \succ B \succ C$	Téo	Bordeaux	$T \succ S \succ Z$
$B \succ A \succ C$	Sandrine	Avignon	$S \succ T \succ Z$

Propriétés de l'algorithme de Gale-Shapley

Observation 1. Les hôpitaux proposent aux étudiants par ordre de préférence décroissante.

Observation 2. Une fois qu'un étudiant est apparié, il ne devient plus jamais non apparié. Il ne fait que changer de choix

Observation 4. La boucle Tant que est itérée au plus $|\mathcal{H}| \cdot |\mathcal{E}|$.

Observation 5. L'ensemble $\mathcal{M}$ retourné par l'algorithme est bien couplage car

- Seul un hôpital non-apparié fait une proposition.
- Un étudiant garde uniquement la meilleur proposition qu'il a reçu.

Propriétés de l'algorithme de Gale-Shapley

Observation 1. Les hôpitaux proposent aux étudiants par ordre de préférence décroissante.

Observation 2. Une fois qu'un étudiant est apparié, il ne devient plus jamais non apparié. Il ne fait que changer de choix

Observation 4. La boucle Tant que est itérée au plus $|\mathcal{H}| \cdot |\mathcal{E}|$.

Observation 5. L'ensemble $\mathcal{M}$ retourné par l'algorithme est bien couplage car

- Seul un hôpital non-apparié fait une proposition.
- Un étudiant garde uniquement la meilleur proposition qu'il a reçu.

Propriétés de l'algorithme de Gale-Shapley

Observation 1. Les hôpitaux proposent aux étudiants par ordre de préférence décroissante.
Observation 2. Une fois qu'un étudiant est apparié, il ne devient plus jamais non apparié. Il ne fait que changer de choix
Observation 4. La boucle Tant que est itérée au plus $|\mathcal{H}| \cdot |\mathcal{E}|$.
Observation 5. L'ensemble $\mathcal{M}$ retourné par l'algorithme est bien couplage car

- Seul un hôpital non-apparié fait une proposition.
- Un étudiant garde uniquement la meilleur proposition qu'il a reçu.

Propriétés de l'algorithme de Gale-Shapley

(Ici on suppose qu'il y a autant d' hôpitaux que d'étudiants. Tous les étudiants postulent à toutes les universités.)

Lemme

Tous les hôpitaux sont appariés dans le couplage $\mathcal{M}$ retourné par l'algorithme.

Corollaire

Tous les étudiants sont appariés.

Corollaire

Tous les étudiants sont appariés.

Propriétés de l'algorithme de Gale-Shapley

(Ici on suppose qu'il y a autant d' hôpitaux que d'étudiants. Tous les étudiants postulent à toutes les universités.)

Lemme

Tous les hôpitaux sont appariés dans le couplage $\mathcal{M}$ retourné par l'algorithme.

Preuve par contradiction :

- Supposons qu'il existe un hôpital **u** non-apparié.
- Il existe aussi un étudiant **s** non-apparié
- **s** n'a jamais reçu de proposition par l'observation 2.
- Par conséquence **u** n'a pas fait de proposition à **s**.
- Comme **u** n'est pas apparié, **u** a du proposé à tous les étudiants (et aussi à **s**).
- D'où la contradiction.

Propriétés de l'algorithme de Gale-Shapley

(Ici on suppose qu'il y a autant d' hôpitaux que d'étudiants. Tous les étudiants postulent à toutes les universités.)

Lemme

Tous les hôpitaux sont appariés dans le couplage $\mathcal{M}$ retourné par l'algorithme.

Preuve par contradiction :

- Supposons qu'il existe un hôpital **u** non-apparié.
- Il existe aussi un étudiant **s** non-apparié comme il y a autant d'hôpitaux que d'étudiants
- **s** n'a jamais reçu de proposition par l'observation 2.
- Par conséquence **u** n'a pas fait de proposition à **s**.
- Comme **u** n'est pas apparié, **u** a du proposé à tous les étudiants (et aussi à **s**).
- D'où la contradiction.

Propriétés de l'algorithme de Gale-Shapley

(Ici on suppose qu'il y a autant d' hôpitaux que d'étudiants. Tous les étudiants postulent à toutes les universités.)

Lemme

Tous les hôpitaux sont appariés dans le couplage $\mathcal{M}$ retourné par l'algorithme.

Preuve par contradiction :

- Supposons qu'il existe un hôpital **u** non-apparié.
- Il existe aussi un étudiant **s** non-apparié
- **s** n'a jamais reçu de proposition par l'observation 2.
- Par conséquence **u** n'a pas fait de proposition à **s**.
- Comme **u** n'est pas apparié, **u** a du proposé à tous les étudiants (et aussi à **s**).
- D'où la contradiction.

Propriétés de l'algorithme de Gale-Shapley

(Ici on suppose qu'il y a autant d' hôpitaux que d'étudiants. Tous les étudiants postulent à toutes les universités.)

Lemme

Tous les hôpitaux sont appariés dans le couplage $\mathcal{M}$ retourné par l'algorithme.

Preuve par contradiction :

- Supposons qu'il existe un hôpital **u** non-apparié.
- Il existe aussi un étudiant **s** non-apparié
- **s** n'a jamais reçu de proposition par l'observation 2.
- Par conséquence **u** n'a pas fait de proposition à **s**.
- Comme **u** n'est pas apparié, **u** a du proposé à tous les étudiants (et aussi à **s**).
- D'où la contradiction.

Propriétés de l'algorithme de Gale-Shapley

(Ici on suppose qu'il y a autant d' hôpitaux que d'étudiants. Tous les étudiants postulent à toutes les universités.)

Lemme

Tous les hôpitaux sont appariés dans le couplage $\mathcal{M}$ retourné par l'algorithme.

Preuve par contradiction :

- Supposons qu'il existe un hôpital **u** non-apparié.
- Il existe aussi un étudiant **s** non-apparié
- **s** n'a jamais reçu de proposition par l'observation 2.
- Par conséquence **u** n'a pas fait de proposition à **s**.
- Comme **u** n'est pas apparié, **u** a du proposé à tous les étudiants (et aussi à **s**).
- D'où la contradiction.

L'algorithme de Gale-Shapley

Lemme

Le couplage $\mathcal{M}$ retourné par l'algorithme n'admet pas de paire bloquante

Preuve : Soit $(\mathbf{u}, \mathbf{s})$ qui ne sont pas dans $\mathcal{M}$.

Cas 1 : $\mathbf{u}$ n'a jamais fait de proposition à $\mathbf{s}$.

Cas 2 : $\mathbf{u}$ a fait une proposition à $\mathbf{s}$.

L'algorithme de Gale-Shapley

Lemme

Le couplage $\mathcal{M}$ retourné par l'algorithme n'admet pas de paire bloquante

Preuve : Soit $(\mathbf{u}, \mathbf{s})$ qui ne sont pas dans $\mathcal{M}$.

Cas 1 : **u** n'a jamais fait de proposition à **s**.

Cas 2 : **u** a fait une proposition à **s**.

Donc si dans les deux cas, $(\mathbf{u}, \mathbf{s})$ n'est pas une paire bloquante de $\mathcal{M}$, alors le lemme est prouvé.

L'algorithme de Gale-Shapley

Lemme

Le couplage $\mathcal{M}$ retourné par l'algorithme n'admet pas de paire bloquante

Preuve : Soit $(\mathbf{u}, \mathbf{s})$ qui ne sont pas dans $\mathcal{M}$.

Cas 1 : **u** n'a jamais fait de proposition à **s**.

u préfère $\mathcal{M}(\mathbf{u})$ à **s**.

$(\mathbf{u}, \mathbf{s})$ n'est pas une paire bloquante de $\mathcal{M}$

Cas 2 : **u** a fait une proposition à **s**.

L'algorithme de Gale-Shapley

Lemme

Le couplage $\mathcal{M}$ retourné par l'algorithme n'admet pas de paire bloquante

Preuve : Soit $(\mathbf{u}, \mathbf{s})$ qui ne sont pas dans $\mathcal{M}$.

Cas 1 : **u** n'a jamais fait de proposition à **s**.

u préfère $\mathcal{M}(\mathbf{u})$ à **s**.
$(\mathbf{u}, \mathbf{s})$ n'est pas une paire bloquante de $\mathcal{M}$

Cas 2 : **u** a fait une proposition à **s**.

s a rejeté la proposition **u**
s préfère $\mathcal{M}(\mathbf{s})$ à **u**.
$(\mathbf{u}, \mathbf{s})$ n'est pas une paire bloquante de $\mathcal{M}$

L'algorithme de Gale-Shapley

Théorème [Gale-Shapley 1962]

L'algorithme de Gale-Shapley retourne une affectation stable pour n'importe quelle instance du problème.

Remarque : Il y a plusieurs applications liées aux problèmes de marriage stable.

Plan

- Les graphes : un outil pour la modélisation
- Terminologie sur les graphes
- Les graphes classiques
  - Les graphes planaires
  - Graphes bipartis
  - Les graphes complets
  - Les arbres
  - Structure de graphes
- Représentation des graphes
  - Matrice d'adjacences
  - Liste de successeurs
- Affectation d'étudiants dans les hôpitaux
  - Description & Formalisation
  - Quel type d'objects il faut calculer ?
  - Extension : et si les acteurs avaient des préférences
- Conclusion

Thèmes abordés pendant cette séance.

- Graphes : notation + représentation
- Comment modéliser un problème :
  - affectations des étudiants
  - utilisation d'algorithmes classiques
- Trouver des instances qui permettent de mettre à défaut l'utilisateur.
- Complexité en terme de mémoire et en terme d'opérations

La semaine suivante les algorithmes gloutons