Filière (DUT) : Développement Web et Multimédia/S1

Département : Génie Informatique

Polycopié de cours :

Théorie de graphes et Théorie de jeux

Préparé par :

Idriss CHANA

Enseignant chercheur au Département GI

EST-UMI Meknès

i.chana@umi.ac.ma

Au Profit des Etudiants de La Première Année de la filière DWM inscrits en semestre S1

Années Universitaires : 2020/2021 à 2023/2024

Table des matières

Introduction

La théorie des graphes et la théorie des jeux sont deux branches fascinantes des mathématiques et de la science de la décision qui trouvent des applications dans une multitude de domaines, allant de l'informatique à l'économie en passant par la biologie et la sociologie. Ce cours introductif vous emmènera dans le monde des graphes et des jeux, en vous fournissant des bases solides pour comprendre et résoudre divers problèmes pratiques.

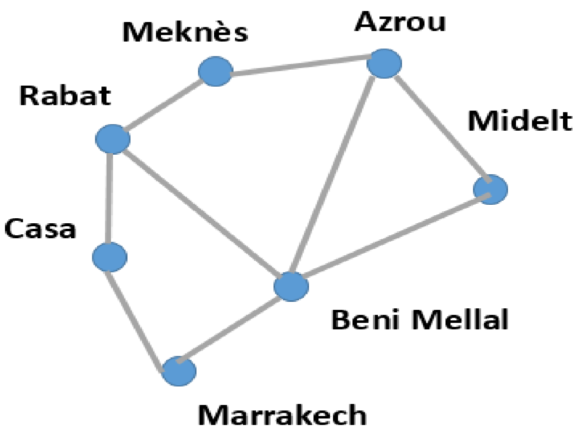
Partie I) Théorie de Graphes

Chapitre 1 : Introduction aux Graphes

1 Introduction

La résolution de plusieurs problèmes nécessite le traçage des petits points reliés par des lignes deux à deux formant ainsi un dessin, ce dernier est nommé **Graphe,** où les points sont les **sommets** et les lignes sont des **arcs ou arêtes,** selon que la relation entre deux point est **orienté** ou **non orienté**.

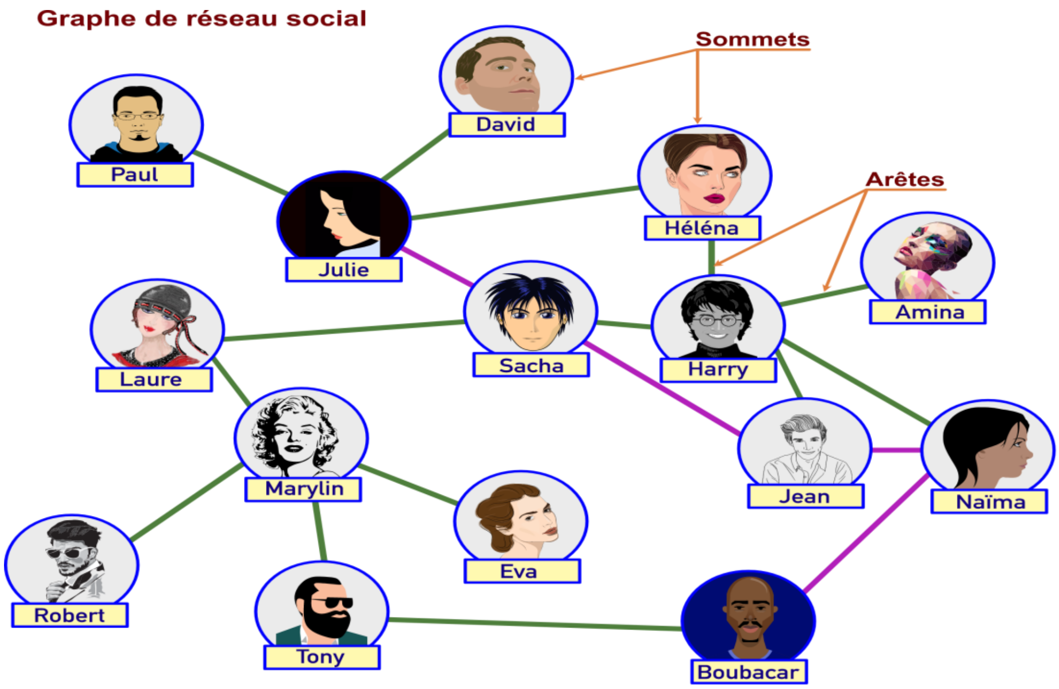
1.1 Exemple 1 : Graphe reseau routier



A partir de ce graphe on peut tirer des informations comme :

- Le plus court chemin entre Meknès et Marrakech
- Existe-t-il un chemin entre deux villes données ?
- Etc……

1.2 Exemple 2 : Graphe reseau Social



A partir de ce graphe on peut tirer des informations comme :

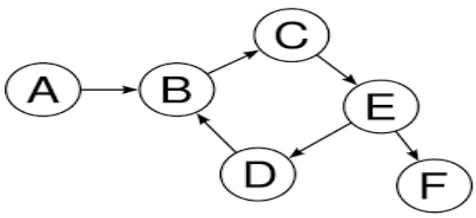
- Quels sont les amis de Harry ?
- Quels sont les amis à proposer à Paul ?
- …..

2 Formalisme

2.1 Définition 1 :

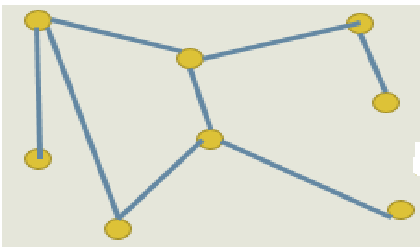
Un graphe noté **G = (S, A)** est un couple formé de deux ensembles :

- Ensemble **S = {s1, s2,…..,sn}** dont les éléments sont appelés **sommets**,
- Ensemble A = {a1, a2…, am} est un **Ensemble de couple** de sommets ***{si,sj}*** ***(Arc ou arrêtes selon la nature de graphe*** qui peut être **orienté ou non orienté**



Graphe Orienté est défini par

Sommets et arcs



Graphe Non Orienté est défini par

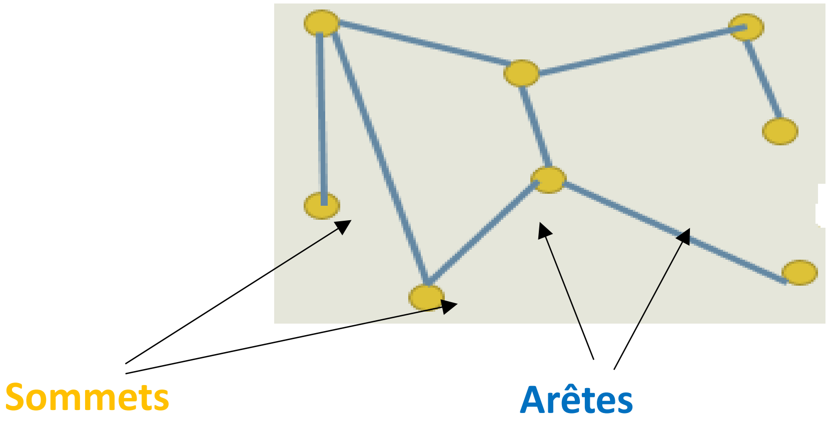
Sommets et arêtes

2.2 Définition 2 : Graphes non orientés

Un graphe simple noté ***G = (S, A)*** est un couple formé de deux ensembles :

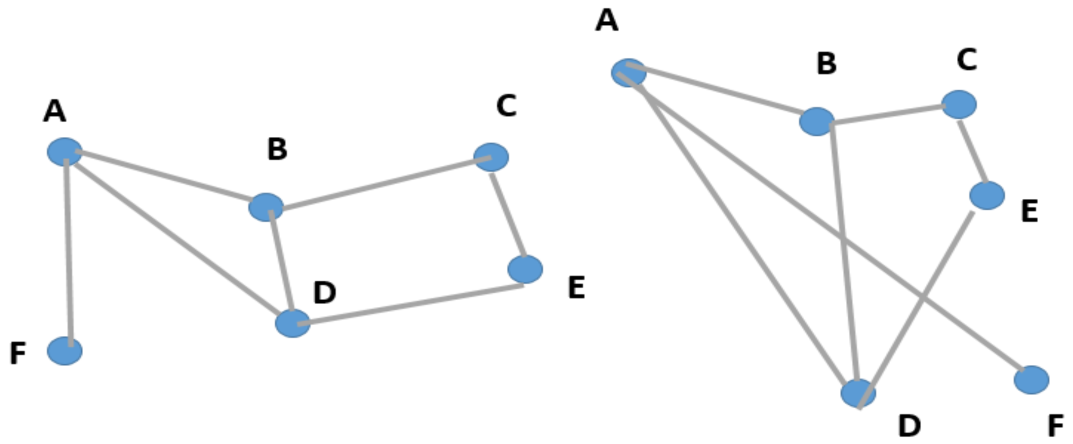
- Ensemble ***S = {s1, s2,…..,sn}*** dont les éléments sont appelés **sommets**,
- Ensemble *A* = ***{a1, a2…,am}*** **partie** à deux éléments **si , sj** de **S**, dont les composantes sont appelés **arêtes**. Avec I **et j** ∊ ***{1, 2,……..n}***

- Lorsque ***a* = {*si, sj*}** appartient à **A,** on dit que **a** est **l'arête** de **G** d'extrémités ***si et sj*,** ou que **a** **joint** ***si*** *et* ***sj*,** ou que **a** **passe** par **si** et **sj.** Les sommets **si** et ***sj*** sont dits **adjacents** dans ***G*** et **l'arete a** représentée graphiquement par ***si—sj***



2.3 Graphe non orienté Représentations et exemples

Les deux figures ci-dessous représentent le **même graphe**



A partir de ce graphe :

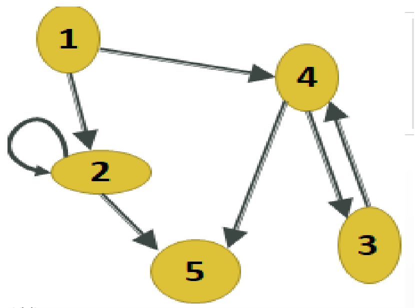
- ✓ Les **sommets** de graphe sont A**, B, C, D, E, F**
- ✓ Les **arêtes** de graphe sont : {**(A, B)** ,**(A,D)** ,**(A, F)** ,**(B,C)** ,**(B,D)** ,**(C,E)** ,**(E,D)**}
- ✓ Exemple de sommets **adjacents :** {**A, B**} **,** {**C,E**} **encore** {**E,D**} **…**
- ✓ Exemples de sommets non **adjacents :** {**E, F**} **.** {**C,D**} **ou encore** {**A,F**} **…**

2.4 Définition : Graphes orientés

Dans un graphe **orienté G(S,A)**, les couples ***(si , sj) ∈A*** sont orientés, c'est à dire que ***(si , sj)*** est un couple ordonné, où ***si*** est le sommet initial, et ***sj*** le sommet terminal.

Un couple ***(si , sj)*** est appelé un **arc**, et est représenté graphiquement par **si → sj** .

Exemple



Le graphe à gauche représente : Le **graphe orienté G = (S, A)** avec

S = {1,2,3,4,5} et **A** = {(1,2), (1,4), (1, 3), (2, 5), (4,3)}.

Les Arcs (si → sj) sont :

$1 \rightarrow 4,\ 1 \rightarrow 2,\ 2 \rightarrow 2,\ 2 \rightarrow 5,\ 3 \leftrightarrow 4,\ 4 \rightarrow 5$

2.4.1 Exercice 1 :

a) Dessiner le graphe non orienté défini par :

G = ({1,2,3,4,5}, {(1,2), (1,4), (1,3), (2,3), (2,4), (4,5)})

b) Dessiner le graphe orienté défini par

GO= ({1,2,3,4,5}, {(1,2), (1,3), (3,2), (3,4), (4,5)}

Chapitre 2 : Terminologies

Dans le domaine passionnant de la théorie des graphes, la compréhension des différentes notions et terminologies est essentielle pour explorer et analyser diverses structures et relations complexes. Dans cette section, nous allons plonger dans les concepts fondamentaux tels **que l'adjacence, l'Incidence, les successeurs, les prédécesseurs, l'ordre, la boucle, le graphe simple**, **le graphe partiel, le sous-graphe, Graphe élémentaire, Graphe complet, le Graphe Valué et le degré d'un sommet.** Ces termes sont des pierres angulaires de la théorie des graphes et nous serviront de bases solides pour aborder des problèmes plus avancés et captivants. Commençons donc par définir ces concepts de base et explorer leur importance dans l'étude des graphes.

1 Terminologies : Adjacence, Incidence, successeurs, prédécesseurs

a) Dans un graphe **non orienté**, un **sommet *si*** est dit **adjacent** au **sommet *sj***

S'il existe une **arête *e=(si ,sj).***

L'ensemble des sommets adjacents à un sommet ***si*** est défini par :

$$adj(si) = \{sj \,/\, (si\,,\, sj\,) \in A\ \}$$

D'où si on a une **arête e=(si,sj),** on dit que:

- ✓ ***Si et sj*** sont **adjacents.**
- ✓ ***Si et sj*** sont **incidents** à l'arête **e.**
- ✓ **e** est **incidente** à ***si et sj***.
- ✓ Deux arêtes sont **adjacentes** si elles sont **incidentes** à un même sommet.

✓ Si $si=sj$, l'arête **e** est une boucle.

b) Dans un graphe **orienté**, on distingue les sommets **successeurs** des sommets **prédécesseurs** :

a. **succ(si)** = {sj/(si , sj) ∈ A}

b. **pred(si)** = {sj/(sj , si) ∈ A}

2 Terminologies : Ordre, boucle, graphe simple, graphe partiel, sous graphe

- ✓ **L'ordre** d'un graphe est le nombre de ses sommets.
- ✓ **Une boucle** est un arc ou une arête reliant un sommet à lui-même.
- ✓ Un graphe non-orienté est dit **simple** s'il **ne comporte pas de boucle**, et **s'il ne comporte jamais plus d'une seule arête entre deux sommets.**
- ✓ Un graphe non orienté **G (S, A)** qui **n'est pas simple** est un **multi-graphe**. Dans le cas d'un multi-graphe, **A** n'est plus un ensemble mais un multi-ensemble d'arêtes.
- ✓ Un graphe orienté est un **p-graphe** s'il comporte au plus **p arcs** entre deux sommets. Le plus souvent, on étudiera des **1-graphes.**
- ✓ Un **graphe partiel** d'un graphe orienté ou non est le graphe obtenu **en supprimant certains arcs ou arêtes**.
- ✓ Un **sous-graphe** d'un graphe orienté ou non est le graphe obtenu en supprimant **certains sommets et tous les arcs ou arêtes incidents aux sommets supprimés**.

3 Terminologies : Graphe élémentaire, Graphe complet, Graphe Valué

- ✓ Un graphe orienté est dit **élémentaire** s'il ne contient pas de **boucle**.
- ✓ Un graphe orienté est dit **complet, si** pour tout couple de sommets différents

  si, sj $\in S^2$. Il y a un **arc (si, sj)** et un **arc** (sj, si).
- ✓ De même, un graphe non-orienté est dit **complet**, pour toute paire de sommets différents

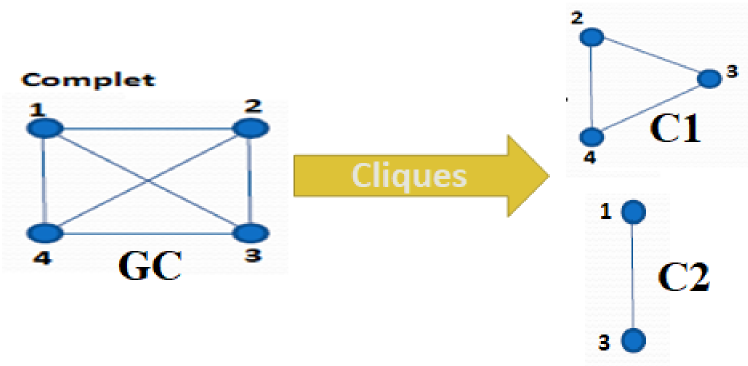
  si,sj $\in S^2$ il y a une arête **(si , sj).**
- ✓ **Stable** : Etant donné un graphe non orienté **G = (S, A),** un stable est un sous-ensemble de sommets S' ⊆ S non connectés par des arêtes de sorte que :

  $\forall (i, j) \in S' \times S' \ , i \neq j \Rightarrow (i, j) \notin A.$

  Autrement Un **stable** est un sous-graphe sans arête.
- ✓ Une **clique** est un sous-graphe complet.

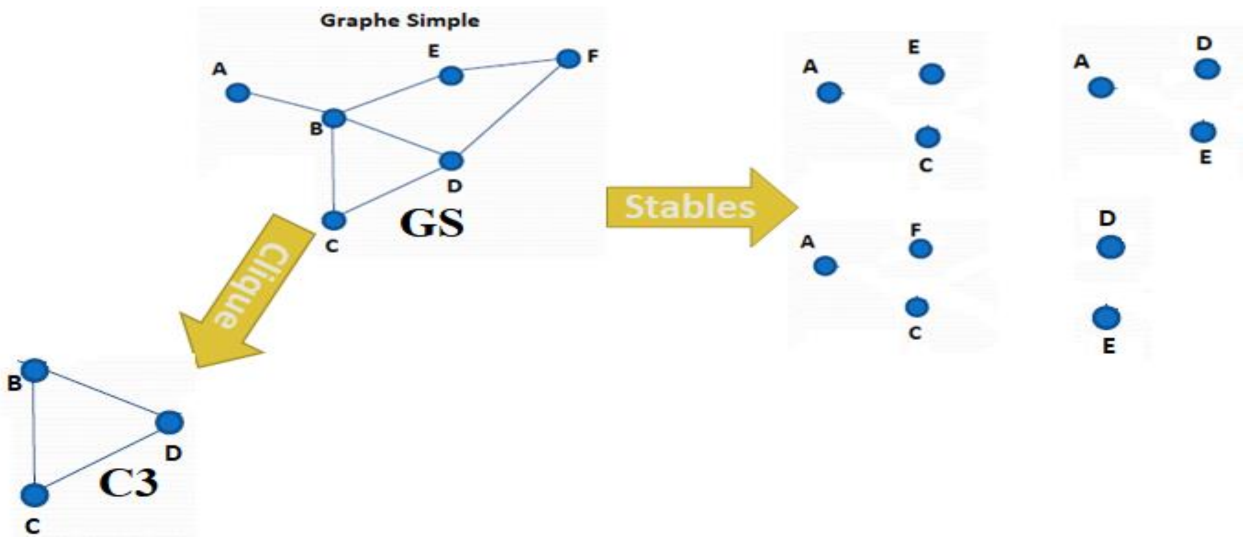
- ✓ Un **graphe valué** est un graphe dans lequel chacune des arêtes présente une **valeur**. Le graphe valué peut être **Orienté ou non**.

Exemple illustratif :

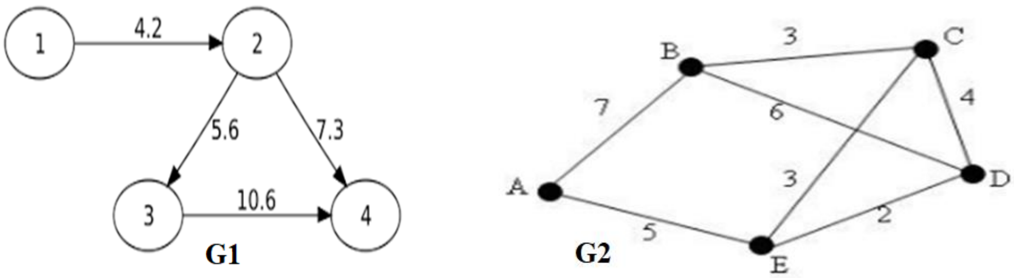


- **A partir de graphe CG ci-dessus qui est un graphe complet (car** pour tout couple de sommets différents il existe **une arrête)** on a extrait à titre d'exemple deux clique **C1 et C2.** Vous pouvez vous en extraire d'autres bien sur

- **Et à partir du graphe GS ci-dessous qui est un graphe simple (**puisque ne comporte pas de boucle, et ne comporte qu'une **seule** arête au plus entre deux sommets) on a extrait à titre d'exemple des **Graphes stables (A,E,C) (A,D, E) (A ,F ,C) et (D, E)** et une **clique C3.** Vous pouvez vous en extraire d'autres bien sur



- Les deux graphes ci-dessous sont des graphes valués (ou dit pondérés)



-

- Dans G1 qui est un **graphe orienté et pondéré(valué):**
    - Le déplacement de **1 à 2** coute **4,2**
    - Le déplacement de **3 à 4** coute **10,6**
    - Le déplacement de **1 à 3** coute **4,2 + 5,6= 9,8**
    - Le déplacement de **1 à 4** coute **4,2 + 7n5= 11,5**
    - **……..**
- Dans **G2** qui est un **graphe no orienté et pondéré(valué):**
    - Le chemin de A à B coute **7**
    - Le chemin de A à D coute **7 +3+4 =14 ou bien 7+6=13**
    - ….

4 Terminologies : Degré d'un sommet

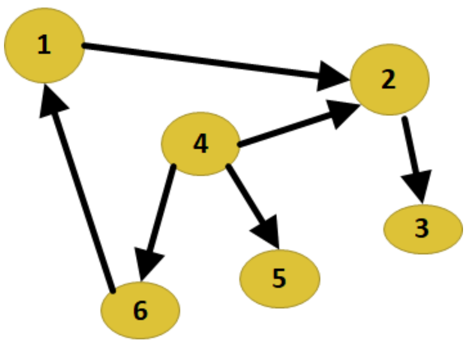
✓ Dans un **graphe non orienté**, **le degré** d'un sommet est le nombre d'arêtes incidentes à ce sommet (dans le cas d'un graphe simple, on aura **d (si) =| adj(si) |.**

✓ Dans un **graphe orienté** :

- Le **demi-degré extérieur** d'un sommet ***si*** , noté **$d°^{+}$(si),** est le nombre d'arcs **partant** de ***si*** (dans le cas d'un 1-graphe, on aura **$d°^{+}$(si) =| succ(si) |.**
- De même, le **demi-degré intérieur** d'un sommet **si** , noté **$d°^{-}$(si),** est le nombre d'arcs **arrivant** à ***si*** (dans le cas d'un 1-graphe, on aura **$d°^{-}$(si) =|pred(si) |.**

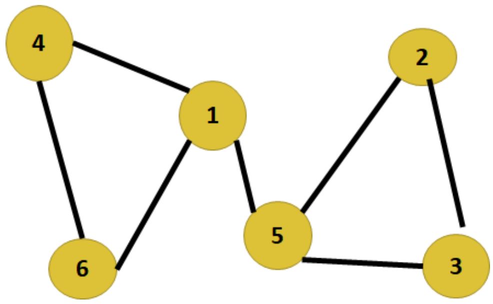
4.1.1 *Exercice 2 :*

Soit le graphe orienté ci-dessous Calculer la somme de tous les **demi-degrés extérieurs** et celle de tous **les demis degrés intérieurs**, et la somme de tous les degrés



4.1.2 Exercice 3 :

Soit le graphe non orienté ci-dessous. Calculer la somme de tous les degrés.



4.1.3 Exercice 4

On considère un graphe orienté G et un graphe non orienté G'

G = (S, A) tel que S = {1, 2, 3, 4, 5} A = {(1, 2),(1, 4),(2, 2),(2, 3),(2, 4),(3, 5),(4, 3),(5, 3)}

G' = (S', A') tel que S' = {A, B, C, D, E} A' = {(A, B),(A, D),(B, C),(B, D),(C, E),(D, E)}

1. Représenter graphiquement les graphes **G** et **G'**
2. Donner le demi-degré extérieur de sommet 2 et le demi-degré intérieur de sommet 4,
3. Donner les sommets prédécesseurs de 4 et les sommets successeurs de 2,
4. Donner un graphe partiel et un sous-graphe de **G et G'**
5. Donner deux **cliques** et deux **stables** de **G'**

4.1.4 Exercice 5 :

Au cours d'une soirée, les invités se serrent les mains les uns les autres (jamais plusieurs fois avec la même personne). Chacun se souvient du nombre de mains qu'il a serrées.

1. Montrer qu'il y a au moins 2 personnes ayant serré le même nombre de mains.
2. Montrer que le nombre total de mains serrées est pair.
3. En déduire que le nombre de personnes ayant serré un nombre impair de mains est pair.

Correction de l'exercice 5

Correction : on construit le graphe non orienté **G = (S, A),** où **S** associe un sommet à chaque convive, et **A** associe une arête à chaque couple de convives qui se sont serrés la main. Le nombre de mains serrées par une personne correspond alors au degré du sommet correspondant dans le graphe.

1. Montrer qu'il y a au moins 2 personnes ayant serré le même nombre de mains revient à montrer qu'il y a au moins 2 sommets ayant le même degré : S'il y a n sommets, le degré d'un sommet est compris entre 0 (cas où le sommet est isolé, c'est à dire que la personne correspondante n'a serré la main à personne) et n − 1 (cas où le sommet est relié à tous les autres, c'est à dire que la personne correspondante a serré la main à toutes les autres). Pour que tous les sommets aient un degré différent, il faut donc qu'il y ait exactement un sommet de degré 0, un sommet de degré 1, ... etc ... et un sommet de degré n − 1. Or, s'il y a un sommet de degré n − 1, il ne peut pas y avoir de sommet de degré 0.

2. Montrer que le nombre total de mains serrées est pair revient à montrer que la somme de tous les degrés est paire : chaque arête ajoute 1 au degré de 2 sommets. Par conséquent, la somme des degrés est X si∈S d ◦ (si) = 2 | A |

3. Montrer que le nombre de personnes ayant serré un nombre impair de mains est pair revient à montrer que le nombre de sommets de degré impair est pair :

On partitionne l'ensemble S des sommets en l'ensemble Spairs des sommets de degré pair et l'ensemble Simpairs des sommets de degré impair. On a X si∈S d ◦ (si) = X sj∈Spairs d ◦ (sj) + X sk∈Simpairs d ◦ (sk) Etant donné que P si∈S d ◦ (si) est pair, et que P sj∈Spairs d ◦ (sj) est pair, on en déduit que P sk∈Simpairs d ◦ (sk) doit aussi être pair. Par conséquent, Simpairs contient un nombre pair de sommets. De façon plus générale, on retiendra que, pour tout graphe simple non orienté, – il existe au moins deux sommets du graphe ayant un même degré ; – la somme des degrés de tous les sommets du graphe est paire et est égale à deux fois le nombre d'arêtes ; – il y a un nombre pair de sommets qui ont un degré impair.

5 Degré d'un sommet : Résultat

De façon plus générale, on retiendra d'après les démonstrations ci-dessus que, pour tout graphe simple non orienté :

- Il existe au moins deux sommets du graphe ayant **le même degré** ;
- La somme des degrés de tous les sommets du graphe est paire et est égale à deux fois le nombre d'arêtes ;
- Il y a un nombre pair de sommets qui ont un degré impair.

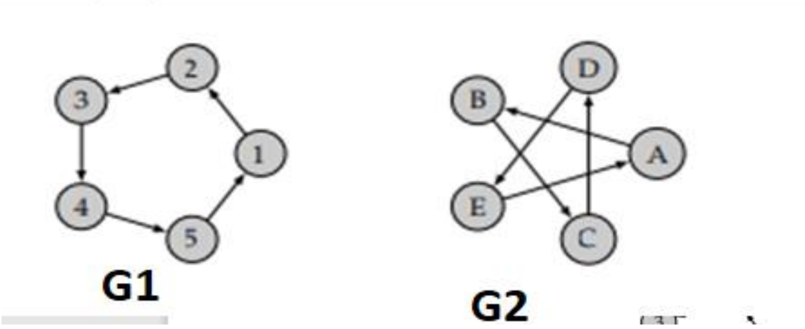
6 Terminologies ; Isomorphisme de graphe :

Deux graphes ***G1 = (S1, A1) et G2 = (S2, A2)*** sont **isomorphes** s'il existe une application bijective ***ϕ : S1 → S2 telle*** que pour tout ***u,v ∈S1 on a (u,v) ∈A ⟺ (ϕ(u), ϕ(v)) ∈A2***.

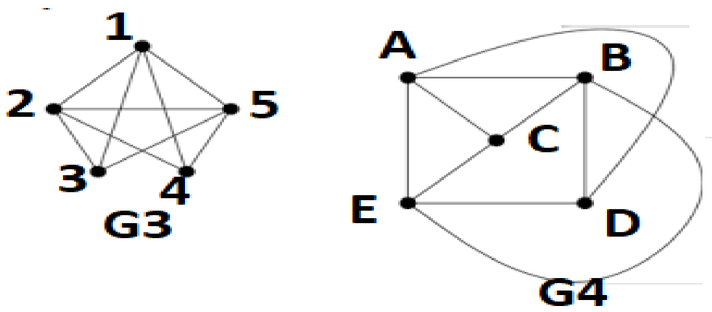
L'application ϕ est alors un **isomorphisme** de graphes.

Exemple 1. Les deux **graphes orientés G1 et G2** suivants sont isomorphes par l'isomorphisme

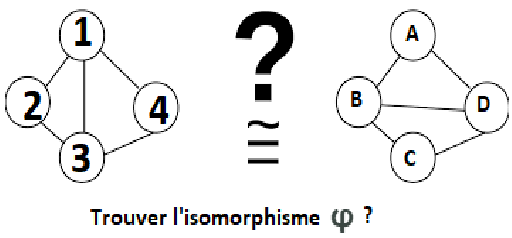
ϕ : 1 → A, 2 → B, 3 → C, 4 → D, 5 → E.



Exemple2. Les deux **graphes non orientés G3 et G4** suivants sont isomorphes par **l'isomorphisme ϕ' : 1 → A, 2 → C, 3 → D, 4 → E, 5 → B.**



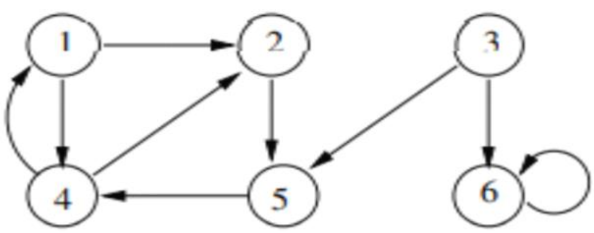
Question : Trouvez l'automorphisme ϕ qui associe les deux graphes ci-dessous



7 Terminologies : Chaîne, cycle, Chemin, circuit, chemin élémentaire et chemin simple

- ✓ Dans un **graphe non orienté**, **une chaîne** est une suite de sommets **<S0, S1, . .,Sn >** telle que chaque deux sommets successives ***Si et S(i+1)*** sont reliés par une arête.
- ✓ **Un cycle** est une chaîne où ***S0 = Sn. (Sommet de départ et le même que le sommet d'arrivée)***
- ✓ Dans le cas **d'un graphe orienté** une **chaîne** est appelée un **chemin** et un **cycle** est appelé **un circuit**.
- ✓ **Un chemin est élémentaire** si les sommets qu'il contient sont tous distincts.
- ✓ **Un chemin est simple** si la séquence d'arcs qui la constitue ne comporte pas des doublons
- ✓ **La longueur d'une chaîne (ou un chemin)** est le nombre d'arêtes qu'elle contient.
- ✓ **Une boucle** est un **circuit** de longueur **1.**

Exemple Considérons par exemple le graphe orienté suivant :



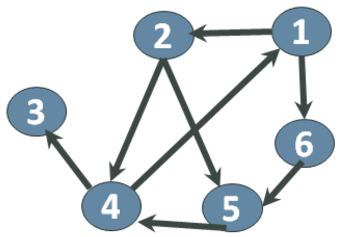
- Un chemin élémentaire dans ce graphe est **< 1, 4, 2, 5 >.**
- Un chemin non élémentaire dans ce graphe est **< 3, 6, 6, 6 >.**
- Un circuit élémentaire dans ce graphe est **< 1, 2, 5, 4, 1 >.**

- Un circuit non élémentaire et non simple dans ce graphe est **< 1, 2, 5, 4, 2, 5, 4, 1 >.**

- Un chemin simple dans ce graphe est **< 1, 4, 1, 2 >** (il ne contient pas d'arcs en double)
- Une boucle < 6,6>

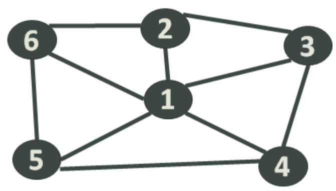
7.1.1 Exercice 6

a) Dans Le graphe orienté ci-dessous



- Donner un chemin qui relie 1 et 4 et un autre qui relie 1 et 3 et déterminer leurs longueurs. Donner un circuit
- Donner un chemin élémentaire, un non élémentaire, un simple et un non simple

b) Dans le graphe Non orienté ci-dessous :



- Donner une chaîne qui relie 1 et 4 et déterminer sa longueur.
- Existe-il un cycle de longueur 5 commençant et terminant par 1

7.1.2 Exercice 7 :

Montrer que s'il existe un chemin/chaine d'un sommet u vers un sommet v dans un graphe, alors il existe un chemin/chaine élémentaire de u vers v

Chapitre 3 Représentation des graphes : Codage par matrices

1 Matrice d'adjacence, liste d'adjacence :

La représentation des graphes par codage est un aspect fondamental de la théorie des graphes, qui est une branche importante des mathématiques et de l'informatique. Elle consiste à décrire comment les relations entre les éléments d'un graphe sont stockées et manipulées dans un format compréhensible par les ordinateurs. Deux des méthodes les plus couramment utilisées pour représenter des graphes sont la matrice d'adjacence et la liste d'adjacence. Ces deux approches ont leurs propres avantages et inconvénients, ce qui les rend adaptées à différentes situations.

a. **Matrice d'adjacence :** La matrice d'adjacence est une représentation tabulaire d'un graphe, souvent utilisée lorsque le graphe est dense, c'est-à-dire qu'il a de nombreuses arêtes. Cette matrice est généralement **une matrice carrée** où les lignes et les colonnes représentent les nœuds (ou sommets) du graphe. Les cellules de la matrice indiquent s'il existe une arête entre deux nœuds. Si l'arête existe, la valeur de la cellule est généralement 1 (ou une autre valeur indiquant le poids de l'arête), sinon, elle est généralement 0.

Avantages de la matrice d'adjacence :

- Représentation efficace pour les graphes denses.
- Accès rapide pour vérifier l'existence d'une arête entre deux nœuds.

Inconvénients :

- Requiert une mémoire considérable pour les graphes de grande taille et peu d'arêtes.
- Les opérations de modification du graphe peuvent être coûteuses.

b. **Liste d'adjacence :** La liste d'adjacence est une représentation plus flexible des graphes, adaptée aux graphes creux (ceux avec peu d'arêtes). Dans cette représentation, chaque nœud du graphe est associé à une liste des nœuds auxquels il est directement connecté (ses voisins). Il peut s'agir d'une liste chaînée, d'un tableau dynamique ou de toute autre structure de données appropriée.

Avantages de la liste d'adjacence :

- Utilise moins de mémoire pour les graphes peu denses.
- Facilite l'ajout et la suppression d'arêtes.

Inconvénients :

- L'accès direct pour vérifier l'existence d'une arête peut être moins efficace que la matrice d'adjacence, en particulier pour les graphes denses.

Le choix entre la matrice d'adjacence et la liste d'adjacence dépend du type de graphe que vous traitez et des opérations que vous prévoyez d'effectuer. Dans de nombreux cas, il peut également être utile de combiner les deux représentations pour tirer parti de leurs avantages respectifs lors de la résolution de problèmes complexes liés aux graphes

1 Matrice d'incidences

La matrice d'incidences est une autre méthode de représentation des graphes, tout comme la matrice d'adjacence et la liste d'adjacence. Cependant, elle est principalement utilisée pour les graphes orientés et les graphes valués, où l'orientation et le poids des arêtes sont essentiels à la représentation.

Voici comment fonctionne la matrice d'incidences :

1. Chaque ligne de la matrice d'incidences correspond à un nœud du graphe.
2. Chaque colonne de la matrice d'incidences correspond à une arête du graphe.
3. Les entrées de la matrice sont remplies de manière à indiquer l'incidence des arêtes sur les nœuds.

Plus précisément :

- Si l'arête se connecte au nœud, l'entrée de la matrice est généralement 1 (ou un autre nombre indiquant le poids de l'arête si le graphe est valué).
- Si l'arête part du nœud, l'entrée est généralement -1.
- Si l'arête n'est pas incidente au nœud, l'entrée est généralement 0.

La matrice d'incidences est donc une représentation compacte des relations entre les nœuds et les arêtes dans un graphe orienté. Elle est particulièrement utile pour résoudre des problèmes tels que la recherche de cycles dans un graphe ou la résolution de problèmes de circulation dans les réseaux.

Avantages de la matrice d'incidences :

- Elle conserve les informations sur l'orientation et le poids des arêtes.
- Elle est utile pour résoudre des problèmes spécifiques liés aux graphes orientés, tels que la recherche de chemins ou de cycles.

Inconvénients :

- Elle peut être plus difficile à interpréter pour des graphes non orientés ou des graphes simples (non valués).
- Elle peut être plus dense et nécessiter plus de mémoire que la liste d'adjacence pour certains types de graphes.

La matrice d'incidences est une autre option précieuse pour représenter et analyser des graphes, mais son utilisation est généralement plus spécialisée et dépend des besoins spécifiques du problème graphique à résoudre.

1.1 Matrice d’incidences : Définition

Soit **G = (S, E)** un graphe non orienté. Supposons que $s_1, s_2, s_3, \dots, s_n$ sont les sommets et $e_1, e_2, \dots, e_m$ sont les arêtes de ***G***.

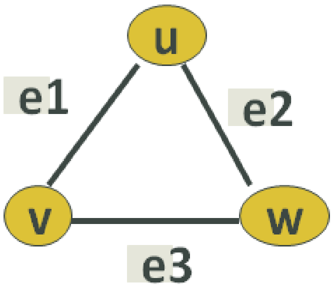
Alors **la matrice d'incidence** par rapport à cet ordre de **S et E** est la matrice de taille **nxm**

(M = **[m_{ij}]) telle** que

$$m_{ij} = \begin{cases} 1 \ si \ \boldsymbol{ei} \ est \ incident \ a \ \boldsymbol{sj} \\ 0 \ si \ non \end{cases}$$

1.2 Matrice d’incidence : Exemple

A partir du graphe ci-dessous



graphe G

Matrice
d'incidence Correspondante au
graphe G

	e_1	e_2	e_3
V	1	0	1
U	1	1	0
W	0	1	1

1.3 Matrice d'adjacence : Définition

Un **graphe** G= **(X, A)** avec N sommets **(|X|=N)** peut être représenté par une **matrice d'adjacence** M, M = **[m_{ij}], de** taille **N*N** telle que :

- **Pour un graphe non orienté**

$$m_{ij} = \begin{cases} 1 \; si \; \{x_i, x_j\} \; est \; une \; arrete \; de \; G \\ 0 \; si \; non \end{cases}$$

- **Pour un graphe orienté**

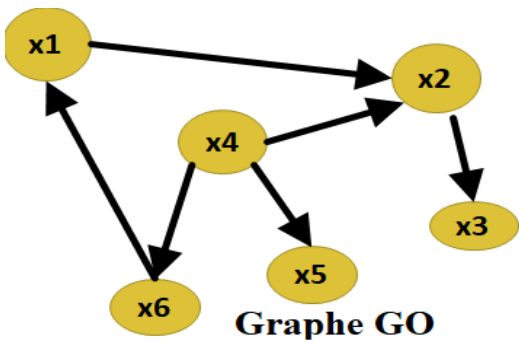
$$m_{ij} = \begin{cases} 1 \; si \; (x_i, x_j) \, est \; un \; arc \; de \; G \\ 0 \; si \; non \end{cases}$$

Exemples :

1.4 Matrice d'adjacence pour un Graphe Orienté

Dans cet exemple nous avons extraire la matrice d'adjacence à partir de graphe Oriente ci-dessous.

Soit **GO** (figure ci-dessous) **un graphe Orienté**.



Graphe GO

En examinant ce graphe, nous extrayons l**a matrice d'adjacence** de suivante

	x1	**x2**	**x3**	**x4**	**x5**	**x6**
x1	0	1	0	0	0	0
x2	0	0	1	0	0	0
x3	0	0	0	0	0	0

x4	0	1	0	0	1	1
x5	0	0	0	0	0	0
x6	1	0	0	0	0	0

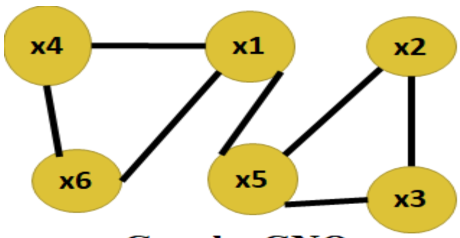
Qu l'on peut exprimer aussi par :

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

1.5 Matrice d'adjacence pour un Graphe Non Orienté

Dans cet exemple nous avons extraire la matrice d'adjacence à partir de graphe **Non Oriente ci-dessous.**

Soit **GNO** (figure ci-dessous) un **graphe Non Orienté**



Graphe GNO

En examinant ce graphe, nous extrayons la matrice d'adjacence de suivante

	x1	**x2**	**x3**	**x4**	**x5**	**x6**
x1	**0**	**0**	**0**	**1**	**1**	**1**
x2	**0**	**0**	**1**	**0**	**1**	**0**
x3	**0**	**1**	**0**	**0**	**1**	**0**
x4	**1**	**0**	**0**	**0**	**0**	**1**
x5	**1**	**1**	**1**	**0**	**0**	**0**
x6	**1**	**0**	**0**	**1**	**0**	**0**

Qu l'on peut exprimer aussi par :

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

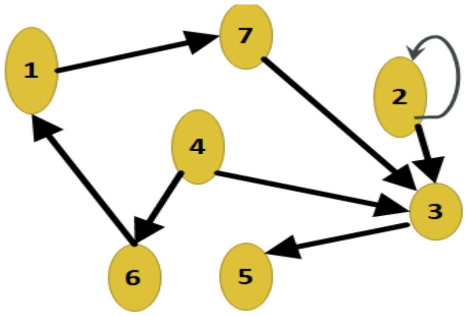
Remarque : La matrice d'adjacences est **une matrice carrée** qui a les propriétés suivantes :

1. Il n'y a que des **zéros 0** sur la diagonale. Un 1 sur la diagonale indiquerait une boucle.

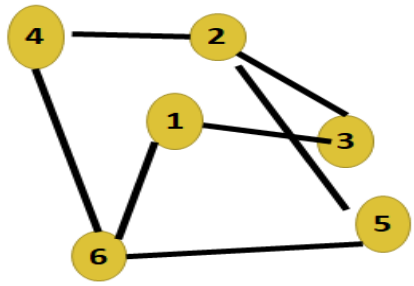
2. Dans le cas non orienté, elle est symétrique : **mij = mji**.

1.5.1 Exercice 8 :

Calculer la matrice d'adjacence des deux graphes GO1 **et GNO1** suivants :



GO1



GNO1

1.5.2 Exercice 9 :

1- Construire un graphe dont la matrice d'adjacence est **M₁** définit comme suit :

$$M_1 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

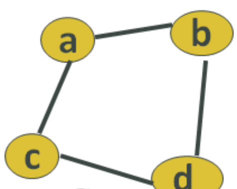
2- Calculer la matrice d'incidence du graphe résultant

a. Matrice d'adjacence : Proposition :

Soit A la matrice d'adjacence d'un graphe G. Le coefficient (u,v) de A^k est le nombre de chaines (chemins) de longueur k qui mènent de u à v.

- **Exemple**

Soit le graphe non orienté G représenté ci-dessous :



G

La matrice d'adjacence de **graphe G** est **A** tel que :

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

Pour trouver les chemins de longueur **4** par exemple, il faut calculer $\mathbf{A^4}$ qui donne :

$$A^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$$

D'où :

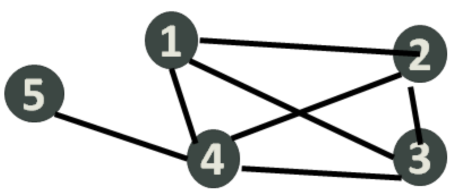
- Le nombre de chemins de longueur **4** de ***a* à *d*** est la (1,4) ème entrée de $\mathbf{A^4}$ **= 8.**
- Le nombre de chemins de longueur 4 de a à b est la $\mathbf{A^4}$ (1,2) = 0.
- Le nombre de chemins de longueur 4 de c à d est la $\mathbf{A^4}$ (4,4) = 8.

Questions :

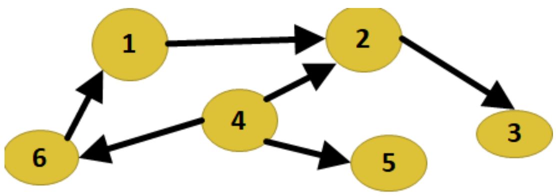
- Quel est Le nombre de chemins de longueur 4 de b à d ?
- Quel est Le nombre de chemins de longueur 4 de b à c ?
- Quel est Le nombre de chemins de longueur 4 de c à d ?

1.5.3 Exercice 10

2- On considère le **graphe non orienté** ci-après Calculer $\mathbf{M^3}$ (M est sa matrice d'adjacence) en déduire le nombre des chaînes de longueur **2** et de longueur **3** qui relie les sommets **1** et **5**



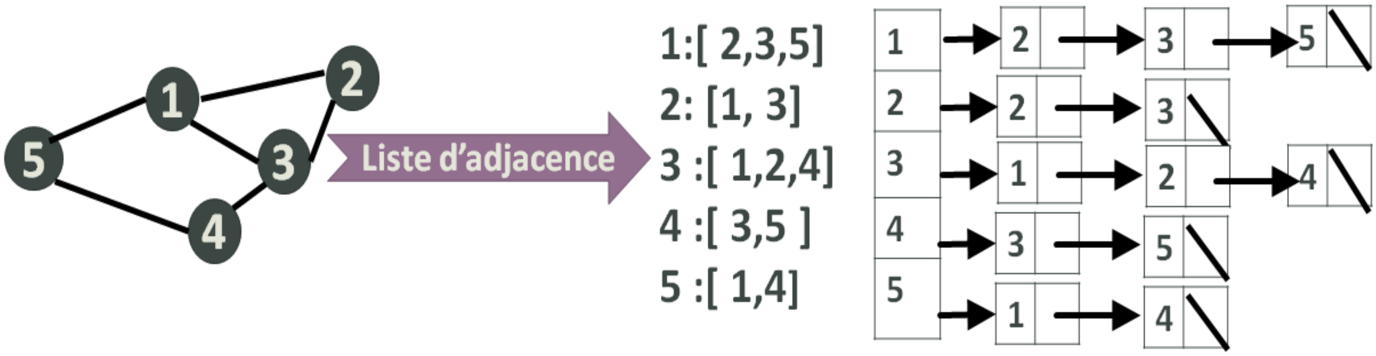
3- Même chose pour le graphe orienté ci-après



2 Liste d'adjacence

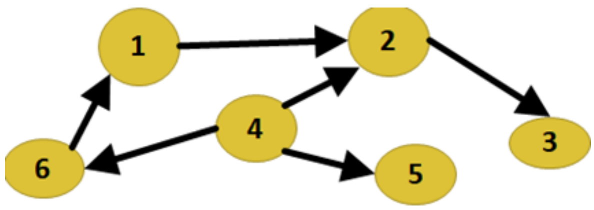
Le codage par **liste d'adjacence** d'un graphe **(S, A)** consiste à associer à chaque sommet ***u*** **∈*S*** la liste des sommet ***v*** tels que *{* ***u,v****}* **∈*A***

Exemple :



2.1.1 Exercice 11 :

- Trouver la liste d'adjacence pour le graphe orienté suivant :



-

Chapitre 4 : Cheminements et connexités :

Ce chapitre explore des concepts fondamentaux de la théorie des graphes qui sont essentiels pour comprendre les structures et les relations entre les éléments d'un réseau ou d'un système complexe. Les graphes, composés de sommets et d'arêtes, sont utilisés pour représenter une grande variété de phénomènes du monde réel, tels que les réseaux sociaux, les systèmes de transport, les réseaux informatiques, et bien d'autres. Dans ce chapitre, nous plongerons dans le concept de cheminement et de connexité en introduisant les notions de **connexité**, c**omposantes connexes**, **composantes Fortement connexes** et **la fermeture transitive**, qui nous permettent de découvrir comment naviguer à travers un graphe, trouver des chemins optimaux, et déterminer si le graphe est connecté ou divisé en plusieurs composantes. Ces concepts jouent un rôle crucial dans de nombreuses applications pratiques, de la planification de trajets dans une ville à la détection de failles dans un réseau informatique

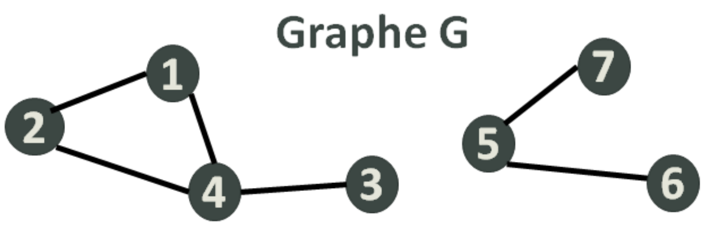
1 Graphe connexe

Définition :

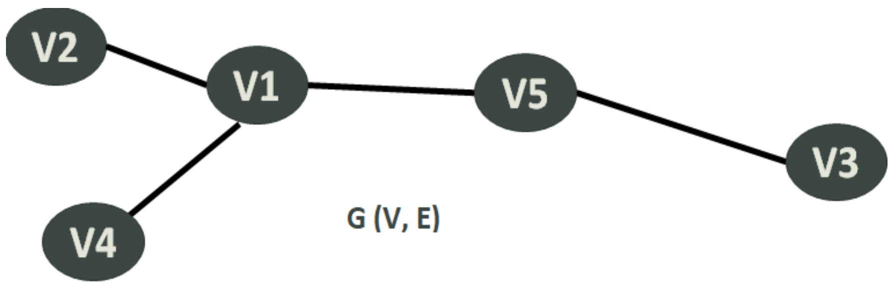
Un graphe G= (S, A) **est connexe** si chaque sommet est accessible à partir de n'importe quel autre. Autrement dit, si pour tout couple de sommets distincts **(u, v)** $\in$ **S²** , il existe une chaine/chemin entre **u et v** .

- **Exemples :**

a) Le graphe **G** suivant n'est **pas connexe**. Car, il n'existe pas de chaîne entre plusieurs sommets : par exemple pas de chaine entre **1** et **5, 2** et **5** , **4** et **7** etc ….



b) Le graphe **G (V, E) ci-dessous est connexe**, **car** pour V = {v_1, v_2, v_3, v_4, v_5}, il **existe toujours** un chemin entre **{v_i, v_j}**, quelque soit **i,j** $\in$**{1,..,5}**



4- Notion d'Articulation et Coupure dans un Graphe

Dans le Cas d'un **Graphe non orienté** on peut définir les notions d'articulation ou dite aussi de coupure comme suit :

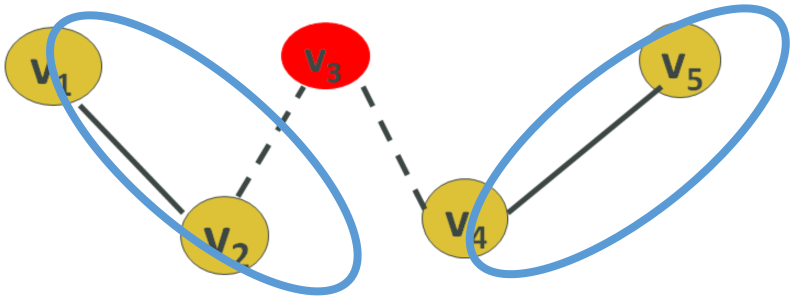
- ✓ **Point d'articulation (sommet de coupure)** : la suppression d'un sommet produit un sous-graphe avec plus de composants connectés que dans le graphe d'origine.
- ✓ La **suppression d'un sommet de coupure** d'un graphe connecté produit un graphe qui n'est pas connecté
- ✓ **Une arête de coupure** : une arête dont la suppression produit un sous-graphe avec plus de composants connectés que dans le graphe d'origine.

- Exemple de point d'articulation (de coupure)

Soit le graphe non orienté ci-dessous :

- **l'arête {v2, v3} est une arête de coupure.**

- le nombre de **composantes connexes selon** le point **de coupure V3** est **2** (> 1)



2 Composante connexe et composante Fortement connexe

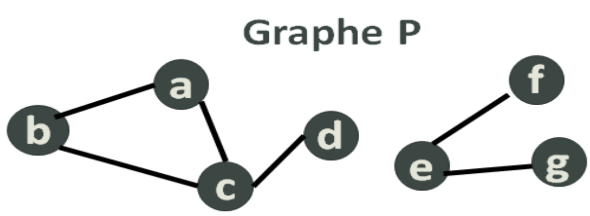
2.1 Composante connexe

Une composante connexe d'un graphe non orienté ***G*** est un sous-graphe **G'** de **G** qui est **connexe et qu'aucun autre sous-graphe** connexe de **G ne contient G'** .

Exemples :

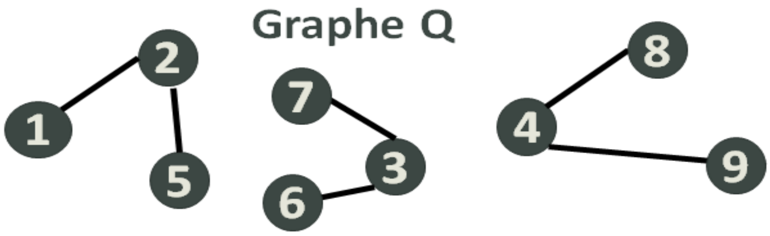
a. Le graphe **P** ci-dessous est composé de 2 **composantes connexes** :

La **première** est le sous-graphe définit par **{a, b, c, d}** et la **seconde** est le sous-graphe définit par **{e, f, g}.**



b. Le graphe **Q** ci-dessous est composé de 3 **composantes connexes** :

La **première** est les sommets **{1, 2, 5}** et la **seconde** est les sommets {**7, 3, 6}** la **3eme** est les sommets {**4, 8, 9**}.



Le nombre de **composantes connexes** est appelé **nombre de connexité** du graphe. Un graphe est connexe Si et seulement si **son nombre de connexité** est égal à 1

2.2 Composante Fortement connexe

Dans le cas des **Graphes Orientés.** On parle de **Graphe et Composante Fortement connexes,** par la suite on utilise les appellations suivantes

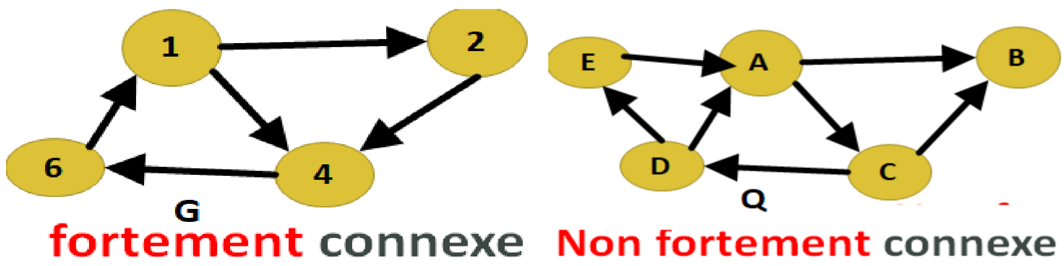
- **Graphe fortement connexe** au lieu de **connexe**,
- **Composante fortement connexe** au lieu de **composante connexe.**

Et qu'on peut définir comme suit :

Un graphe orienté est fortement connexe s'il existe un chemin de ***u*** à **v** et de **v** à **u** pour tous sommets ***u*** et **v** du graphe

Composants fortement connexe : sont des **sous-graphes** d'un graphe qui sont **fortement connexe**

Exemple 1 : graphe orienté fortement connexe

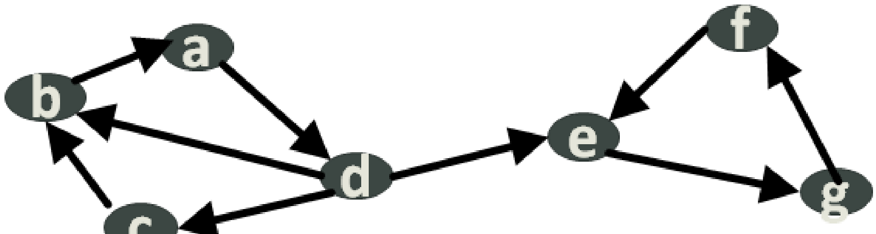


Le graphe **G est** fortement **connexe, car** dans ce graphe on peut aller de n'importe quel sommet vers n'importe quel autre sommet et vice-versa.

Alors que le deuxième graphe Q est **Non Fortement connexe**, car on peut aller d'A vers **B** mais **l'inverse** (B vers **A**) n'**est pas possible**

Exemple 2 : Composants fortement connexe

Soit le graphe G ci-dessous



Graphe G

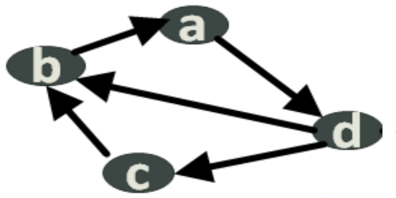
Le graphe **G est-il fortement connexe** ?

Non, il n y a pas de **chemin entre deux sommets quelconque par exemple**

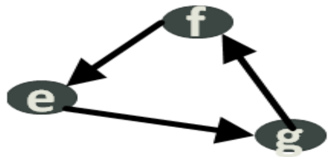
- Pas de chemin de **g** vers **d,**
- Ni de chemin de **f** et **a**

Mais Il Contient **2 composantes fortement connexes qui sont :**

- **Le sous-graphe défini par les sommets {a, b, c, d}**

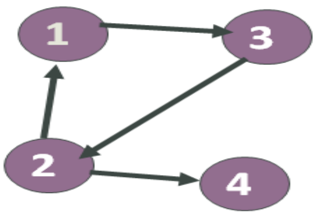


- **le sous-graphe défini par les sommets {e, f, g}.**

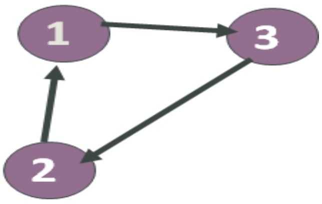


Exemple 3 : Composants fortement connexe

Soit le graphe ci-dessous **n'est pas fortement connexe**



Une composante fortement connexe de ce graphe est le sous graphe suivant **:**



3 Fermeture transitive

La fermeture transitive d'un graphe est un concept important en théorie des graphes qui consiste à étendre les relations directes entre les sommets du graphe pour inclure toutes les relations indirectes possibles. En d'autres termes, elle vise à déterminer quelles paires de sommets sont connectées par un **chemin de longueur quelconque** dans le graphe, en utilisant les arêtes existantes comme base.

3.1 Définition Formelle :

On appelle **fermeture transitive** d'un graphe $G = (S, A)$, le graphe **G+ = (S, A^f)** tel que pour tout couple de sommets $(u\ ,\ v\ ) \in S^2$, l'arc/arête $(u\ ,\ v\ )$ appartient à **A^f si et seulement s'il existe un chemin/chaine de *u* vers *v* .** Autrement dit c'est un graphe qui contient tous les chemins possibles d'un sommet quelconques à un autre sommet quelconque**.**

Explication :

Supposons que nous ayons un graphe (orienté ou non) **G = (V, E),** où V représente l'ensemble des sommets et E l'ensemble des arêtes. La fermeture transitive de ce graphe, généralement notée **G+,** est un nouveau graphe qui a les mêmes sommets que **G (V),** mais dont les arêtes sont définies de manière à inclure toutes les relations transitives possibles entre les sommets.

En d'autres termes, (u, v) est une arête dans **G+** si et seulement s'il existe un chemin dirigé de **u à v** dans le graphe G. Ce chemin peut passer par zéro ou plusieurs autres sommets intermédiaires, mais il doit exister un moyen de se rendre de **u à v** en suivant les arêtes du graphe G.

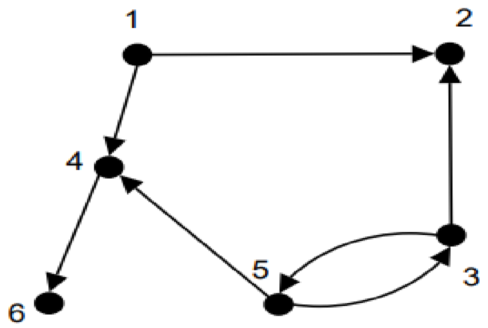
La fermeture transitive d'un graphe est utile dans de nombreuses applications, notamment la détection de relations de dépendance indirectes, la recherche de chemins entre sommets, la modélisation des relations dans les bases de données, et bien d'autres domaines où l'analyse des relations entre les éléments est nécessaire.

3.2 Calcul de la fermeture transitive

Le calcul de la fermeture transitive d'un graphe revient à calculer le nombre de chemins de longueur 2, 3 et n-1 (**n est le nombre de sommets**) qui existent entre deux sommet u, v quelconque Pour se faire on calcul la matrice d'adjacence M du graphe puis ces puissances 2, 3,….n-1 ($\mathbf{M^2\ M^3 \cdots\cdots M^{n-1}}$)

a) Exemple de calcul de la fermeture transitive

Soit le graphe ci-dessous est le graphe pour lequel on se propose de calculer **la fermeture transitive** en calculant les puissances successives des matrices



b) Calculant tout d'abord la matrice M d'adjacence de graphe qui est :

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

c) Calculant en suite successivement ***M^2 , M^3 , M^4 et M^5* (il y a 6 sommets)** représentants respectivement les chemin de longueurs **2, 3 ,4 et 5**

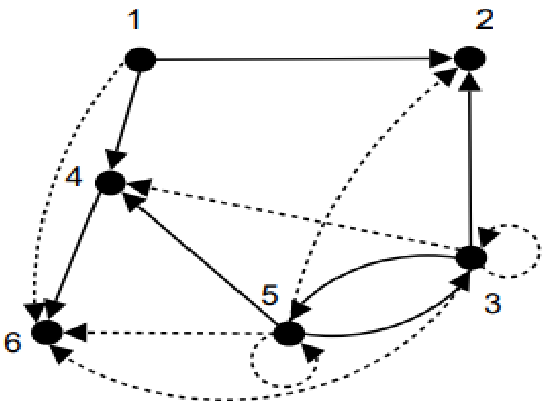
$$M^2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad M^3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$M^4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad M^5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

d) **La disjonction (ou : 0 v 1=1 ; 1 v 1=1 ; 0 v 0 =0)** de ces matrices représente la **matrice de la fermeture transitive** noté **M+**

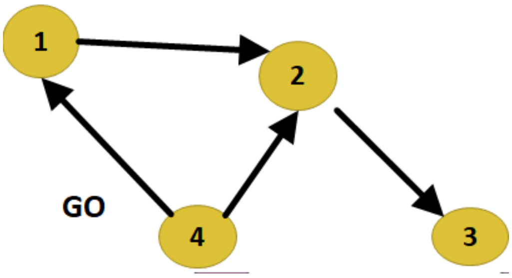
$$M+ = M \vee M^2 \vee M^3 \vee M^4 \vee M^5 = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

e) **Le graphe G+** obtenu de la matrice **M+** de la **fermeture Transitive** est alors le suivant **:**



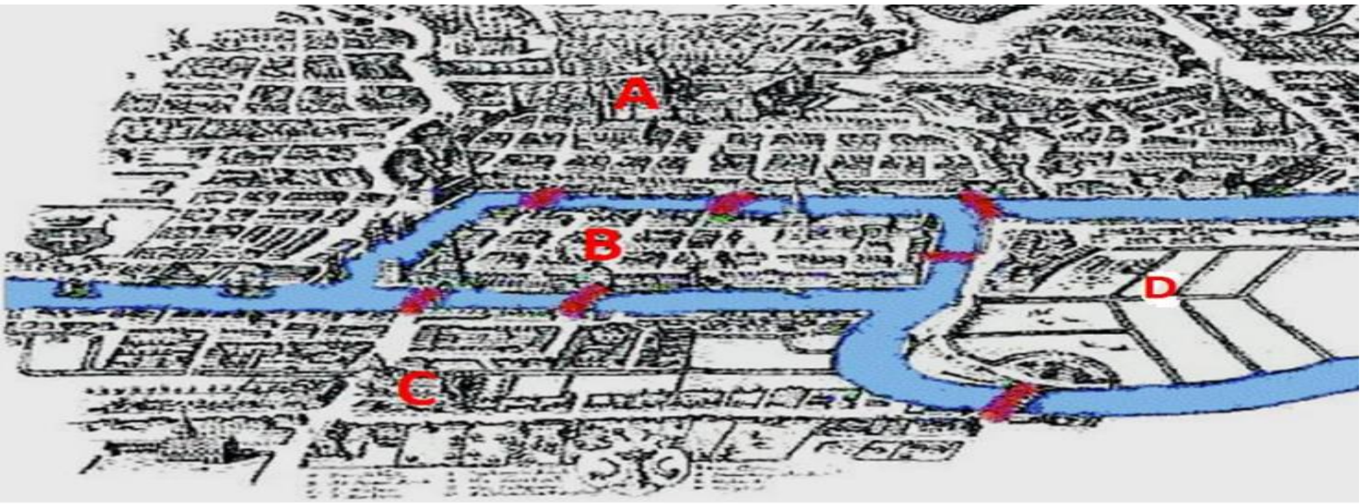
3.2.1 Exercice 12 :

Trouver la fermeture transitive pour le graphe suivante en représentant le graphe **GO+** correspondant :



4 Notion de Graphe Eulérien

Les habitants de **Königsberg**, en Allemagne, se sont demandés s'il était possible de faire une visite à pied de la ville en traversant chacun **des sept ponts** sur la **rivière Presel** une **seule fois**. Est-il possible de commencer par une certain **rive** (nœud) et de faire une promenade en utilisant chaque **pont** (arête) exactement une fois et se termine au **rive** (nœud) de départ ?



4.1 Chaine eulérienne et Cycle Eulérien

4.1.1 Définitions et exemples

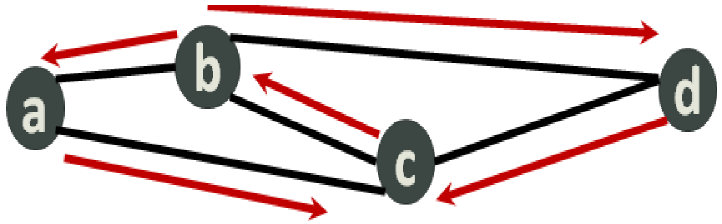
Dans **un graphe non orienté**, une chaine eulérienne est une chaine qui **emprunte une et une seule fois chaque arête du graphe.**

De même, **un cycle eulérien** est un **cycle** *qui* emprunte **une et une seule fois** chaque arête du graphe.

Enfin, **un graphe comportant un cycle eulérien** est appelé **graphe eulérien**.

Exemples :

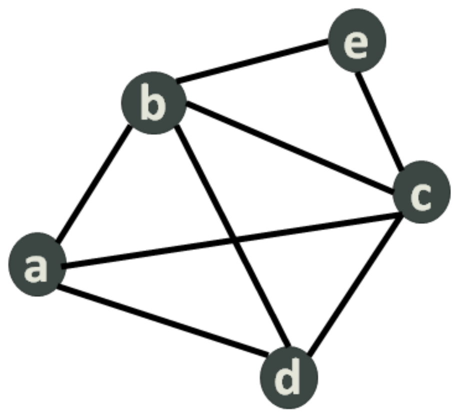
b- Soit le graphe non orienté ci-dessous ou les flèches rouges dessinés sur les arêtes montrent la direction de la chaine pour faciliter la compréhension de l'exemple



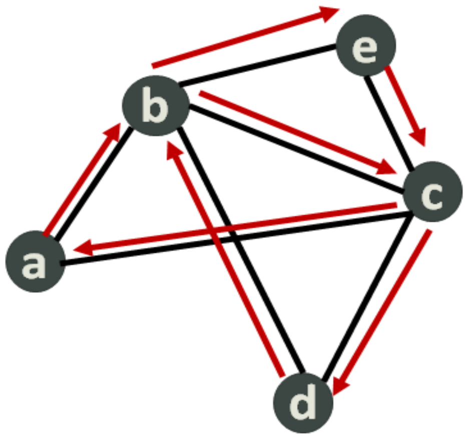
- La chaine **<b,a,c,b,d,c>** est **une chaîne eulérienne,** car en suivant le chemin de cette chaine on peut visiter tous les sommets en traversant **tous les arêtes une** et **une seule fois**
- La chaine **<a,c,d,b,a,c,b>** n'est **pas une chaîne eulérienne,** car on a emprunté l'arête **(a, c) plus qu'une** seule fois (ici deux fois) .

c) Les deux exemples ci-dessous illustrent la notion **de cycle eulérien**

- Le graphe ci-dessous **ne contient pas de cycle eulérien**

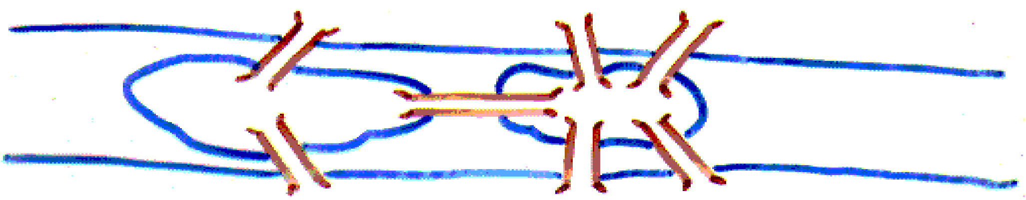


- Tandis que le graphe ci-dessous contient un **cycle eulérien** à savoir **:**
  <a,b,e,c,d,b,c,a>. Donc ce graphe est dit aussi **graphe eulérien**

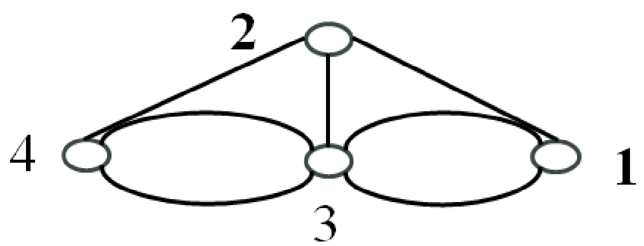


4.2 Théorème d'Euler

Revenons aux sept ponts de **Königsberg** reliant les 4 rives de la rivière **Presel, et** qu'on peut exprimer aussi par le dessin ci-dessous



On peut redessiner l'image d'origine en tant que graphe, en associant une arête à chaque **pont** séparant **les 4 rives** de la ville. Et les **rives** seront représentées dans le graphe **par des sommets**



Selon Euler ce graphe :

- N'a pas de circuit qui utilise chaque arête exactement une fois.
- Même à partir de n'importe quel sommet (**rive**)

Théorèmes d'Euler énoncée (1766)

a. Un graphe ***G*** est ***eulérien*** si et seulement si ***G*** est **connexe** et n'a pas de sommets de **degré impair**

b. Un graphe ***G*** admet une ***chaine/chemin eulérienne*** du sommet ***u*** à un autre sommet ***v*** si et seulement si ***G*** est connexe et $u \neq v$ sont les deux seuls sommets de **degré impair**

- **Preuve ===>**

  Supposons que ***G*** ait un chemin Eulérien ***T*** d'un sommet ***u*** à un autre sommet ***v*** **(u et v pas nécessairement distincts)**. Pour chaque sommet autre que ***u* et *v* , *T*** utilise une arête pour sortir et une autre arête pour entrer. Ainsi, le degré du sommet est pair.

  - Si **u = v** , alors **u** a aussi **un degré pair**=> cycle/circuit **Eulerien**
  - Si **u ǂ v**, alors **u et v** ont tous les **2 un degré impair**. Chemin **d'Eulerien**

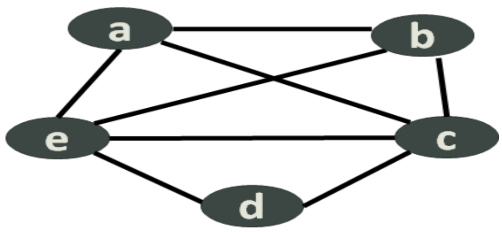
  Preuve <===

  Inversement, soit ***G*** un graphe connexe dont tous les sommets sont de degré pair

  Partant d'un sommet arbitraire *u*, on peut former de proche en proche un cycle élémentaire retournant sur *u* . En effet, chaque fois qu'on emprunte une arête, on la retire du graphe. Quand on aboutit en un sommet *v* ǂ *u*, on peut toujours prolonger la chaine car *v* est de degré pair. Donc, la chaine rencontrera nécessairement le sommet initial *u* sinon on ne s'arrêtera jamais, ou les degrés de *u* et le *dernier sommet* visité sont impairs.

4.2.1 Exercice 13

Montrer que le graphe suivant est eulérien



Correction :

Appliquant le théorème d'Euler

- **Le graphe est connexe et,**
- **Degré (d) = 2, degré (e) = degré (c) = 4** et **degré (a) = degré (b) = 3**. Par conséquent, ce graphe admet **une chaine eulérienne** entre **a et b** (les seuls sommets ayant le degré impair d'après la **partie b du théorème**) **.** Mais, ce graphe n'admet pas de **cycle eulérien** (car il a des sommets de degré impair d'après la **partie a du théorème**)

4.3 L'algorithme qui permet de déterminer un cycle eulérien.

4.3.1 L'algorithme

Une fois on montre qu'un graphe est eulérien en utilisant le théorème d'Euler ci-dessous on applique **l'algorithme** suivant pour déterminer le **cycle eulérien** :

Etape 1)

1.1 Cycle C := un Cycle de graphe **G** commençant à un sommet arbitraire ***v***.

1.2 Ajoutez des arêtes successivement pour former une chaine qui revient à ce sommet (au **sommet de départ v**) .

Etape 2) H := G –C (**H** est obtenu en enlevant tous les sommets **isolés** de cycle C de G)

Etape 3) Boucle itérative : Tant que H a encore des arêtes

3.1 Sous-cycle sc := un cycle qui commence à un sommet de **H** qui est également dans **C**

(dans l'exemple suivant , le sommet « **c** »)

3.2 H := H – sc (- tous les sommets isolés)

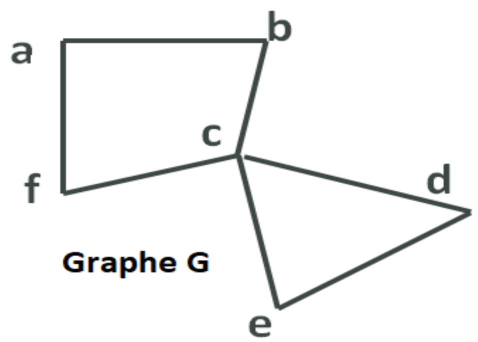
3.3 C := C « réuni» avec sous-circuit sc (autrement C:=C+ sc)

fin tant que

Etape 4) C : est le **Cycle eulérien Obtenu**

4.3.2 Exemple d'application

Nous allons appliquer l'algorithme ci-dessous sur le graphe G suivant pour obtenir un cycle eulérien



Montrons tout d'abord que le graphe **G est Eulérien :**

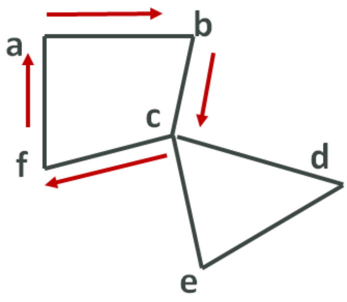
- **G** est connexe et,
- Tous les degrés des sommets de G **sont paire**

Donc le graphe **G** est une Graphe **Eulérien**

Application de l'algorithme

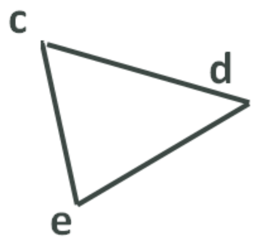
- **Début de l'algorithme**

<u>1)</u> <u>Construisons un cycle simple</u> : {a,b}, {b,c}, {c,f}, {f,a}



- Le **cycle Eulerien est** construit si tout les arêtes sont visités.
- Est-ce vraie dans ce cas ? **Non** car il y a encore **d'autre arêtes non encore visitées**

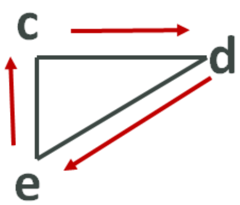
<u>2)</u> <u>Supprimons **la chaine simple**</u> :   **{a,b}, {b,c}, {c,f}, {f,a}**



- ***C*** est le sommet commun entre **ce sous-graph** et le **graphe "parent**".

<u>3)</u> ***C*** a un degré **pair**. Partez de ***C*** et tracez un autre **cycle** :

{c,d}, {d,e}, {e,c}

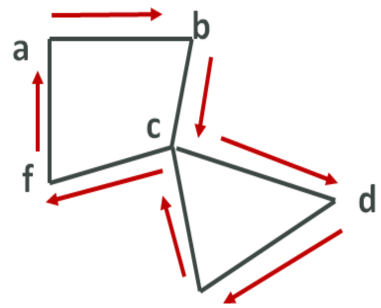


*<u>4)</u> Réunir   les circuits des **2** sous-graphes dans le sommet **comun C :***

{a,b}, {b,c}, {c,f}, {f,a}   "+"   {c,d}, {d,e}, {e,c}

"="

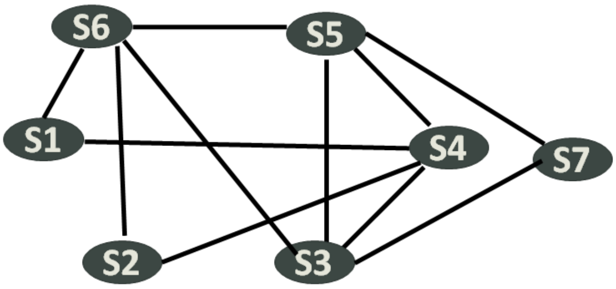
{a,b}, {b,c}, {c,d}, {d,e}, {e,c}, {c,f} {f,a}



- **Fin de l'algorithme**

4.3.3 Exercice 14

1) Montrer que le graphe ci-après admet un cycle eulérien
2) En utilisant l'algorithme précèdent déterminer **un cycle eulérien** pour ce graphe



4.4 L'algorithme qui permet de déterminer une Chaine eulérien.

Rappelons qu'une **chaine eulérien** est une chaine et part d'un sommet et qui permet de traverser tous les arêtes **une et une seule fois sans revenir au sommet de départ**. Dans cette section nous allons présenter un algorithme qui permet de déterminer une **chaine eulérien d'un graphe.**

Pour se faire il faut tout d'abord montrer que le graphe contient cette chaine

- **Algorithme**

Soit G un graphe connexe, admettant une chaine eulérienne.

Etape 1 : Former une chaine reliant les deux sommets de degré impair (en employant une et une seule fois les arêtes)

Etape 2: **Si tous l**es arêtes sont **utilisées** → la **chaine eulérienne est obtenue**

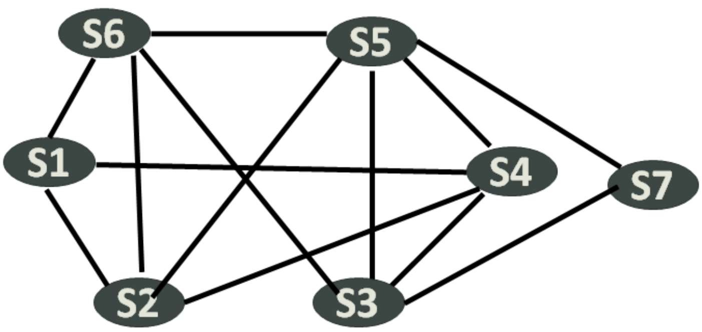
Sinon → choisir un sommet de la chaine formée dans l'étape 1 et former un cycle issu de ce sommet utilisant les arêtes non déjà marquées, et ce une seule fois pour chaque arête.

Etape 3 : retourner à l'étape **2**.

4.4.1 Exercice 15

1) Montrer que le graphe ci-après admet une chaine eulérienne

2) En utilisant l'algorithme de slide précèdent déterminer une chaine eulérienne pour ce graphe

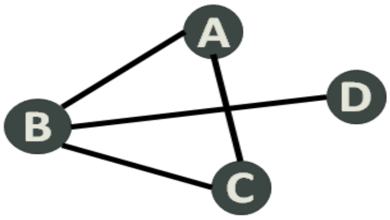


5 Notion de graphe Hamiltonien

5.1 Définition :

Chaîne/chemin hamiltonienne : Chaîne/chemin passant une seule fois par tous les sommets d'un graphe.

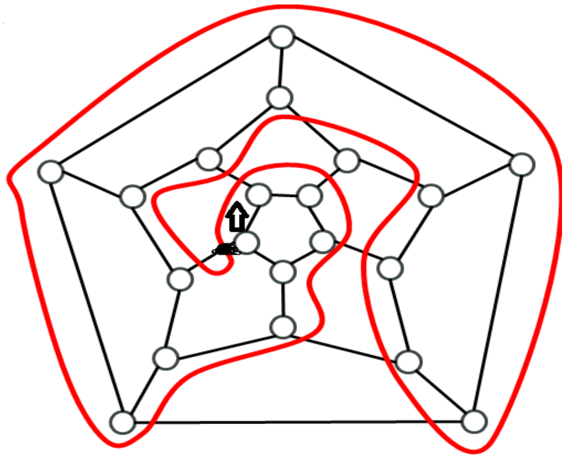
Exemples : DBCA, DBAC,



- **Cycle/Circuit hamiltonien** : passant une seule fois par tous les sommets d'un graphe et revenant au sommet de départ
- Un graphe contenant une **chaine/chemin ou cycle/circuit hamiltonien** est **appelé graphe hamiltonien.**
- **Tout cycle hamiltonien** peut être converti en une **chaine hamiltonien** en supprimant l'une de ses arêtes,
- Mais **une chaine hamiltonien** ne peut être étendu en **cycle hamiltonien** que si ses extrémités sont adjacentes.

Question : Un graphe des sommets d'un dodécaèdre (figure ci-dessous) (Polyèdre à douze faces)

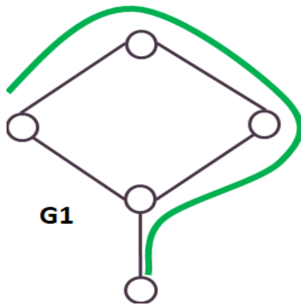
Est-ce hamiltonien ?



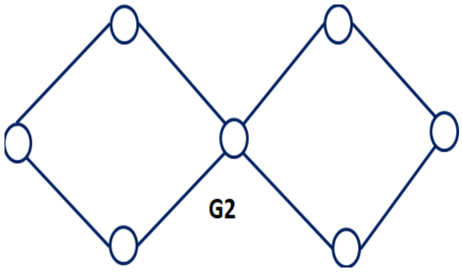
Réponse : Le chemin en couleur tracé autour du graphe partant du sommet qui est à côté de la flèche et revenant à celui-ci visitant tous les sommets une et une seule fois montre que ce graphe est **hamiltonien**

Autres exemples :

- **G1** contient une **chaine Hamiltonienne**, mais pas un cycle Hamiltonien.



- **G2** contient :
    - un cycle Eulerien,
    - Une chaine Hamiltonienne
    - mais il n a pas de cycle Hamiltonienne



Remarque : de nombreux problèmes en recherche opérationnelle consistent à chercher un chemin ou un cycle hamiltonien dans un graphe. Le plus connu est probablement le problème du voyageur de commerce, qui doit trouver un itinéraire “optimal” passant par chaque ville de son réseau commercial. On considère dans ce cas le graphe non orienté valué représentant une carte routière: les sommets correspondent aux villes, les arêtes aux routes, et les valuations aux distances. Il s’agit alors de trouver un **cycle hamiltonien** de valuation minimale. Ce problème est un des premiers à avoir été montré comme étant **NP-complet** (ce qui implique que l’on ne connait aucun algorithme “efficace” pour résoudre ce problème).

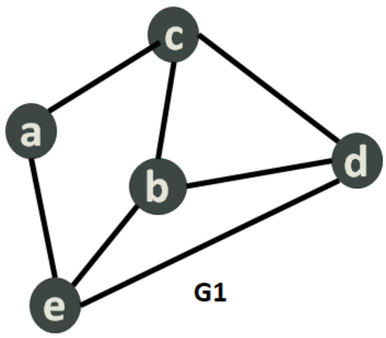
5.2 Graphe Hamiltonien : Théorèmes

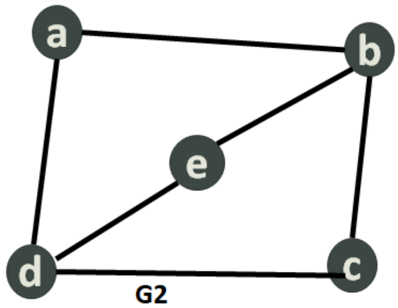
On ne connait aucune condition nécessaire et suffisante d’existance de cycles (chaines, circuits ou chemins) hamiltoniens, valable pour tous les graphes. On peut juste donner des conditions suffisantes, portant en particulier sur les degrés d’un graphe simple.

1. **Théorème de DIRAC** : si **G** est un graphe simple à n sommets avec $n \geq 3$ tel que le degré de chaque sommet de **G** est d'au moins $n/2$ alors G a un **circuit/cycle Hamiltonien**
2. **Théorème d’ ORE** : si G est un graphe simple à n sommets avec $n \geq 3$ tel que **deg (u)** + $deg\ (v) \geq n$ pour chaque paire de sommets **non adjacents** u ***et*** v dans **G**, alors **G** a un **circuit/Cycle Hamiltonien**.

5.2.1 Exercice 16 :

Déterminer une Chaine/Cycle Hamiltonienne pour ces deux graphes ?





Réponse :

Pour **G1 : cycle hamiltonien est <a, e, b, d , c ,a>**

Pour G2 : Ce graphe ne possède **pas de cycle hamiltonien**, mais possède une **chaine hamiltonienne (< a, b, e, d, c >)**

Chapitre 5 : Arbres et arborescence

1 Introduction :

Le chapitre « Arbres et arborescence" explore un domaine fondamental de la théorie des graphes. Il commence par fournir des définitions claires et des propriétés essentielles relatives aux arbres, qui sont des graphes spéciaux sans cycles. Ensuite, il aborde une variété d'exercices pour aider les lecteurs à consolider leur compréhension des concepts et techniques associés aux arbres dans les graphes. Le chapitre se penche également sur les arbres recouvrants, qui sont essentiels pour la connectivité dans les graphes, ainsi que sur les graphes valués, où chaque arête est associée à un poids ou un coût. Une section importante du chapitre se concentre sur l'arbre de poids minimum, qui vise à trouver l'arbre recouvrant ayant la somme minimale des poids d'arêtes. Pour résoudre ce problème, deux algorithmes classiques, l'algorithme de **Prim** et l'algorithme de **Kruskal**, sont présentés en détail, montrant comment trouver efficacement l'arbre de poids minimum dans un graphe valué. Ce chapitre est essentiel pour tout étudiant ou professionnel souhaitant maîtriser les concepts et les outils liés aux arbres et à la recherche d'arbres de poids minimum dans les graphes**.**

2 Définitions, Propriétés, Théorèmes, proposition

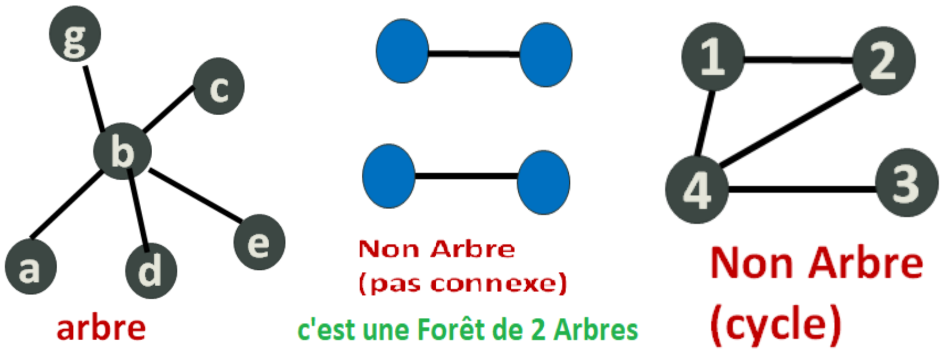
Cette section constitue la base conceptuelle de notre exploration des graphes et de leurs diverses caractéristiques. À travers cette section, nous établirons un socle solide de connaissances en introduisant les termes et concepts fondamentaux qui serviront de fondement à notre étude. En examinant attentivement les définitions, en explorant les propriétés, et en démontrant les théorèmes et propositions essentiels, nous acquerrons une compréhension approfondie des principes sous-jacents à la théorie des graphes, jetant ainsi les bases de nos futures investigations.

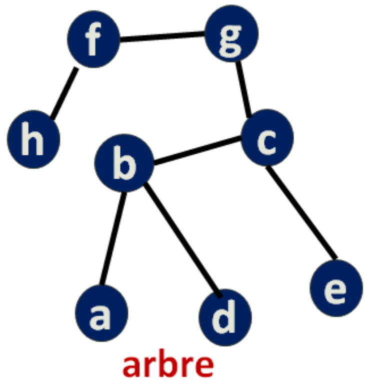
2.1) **Définition 1 : Un** arbre est un graphe connexe sans boucle et sans cycle.

2.2) On appelle **forêt** un graphe dont chaque composante connexe est un arbre.

2.3) On appel **feuil** un sommet de degré 1

Exemples :





- Le sommet **3** est une feuille
- Les sommets **e**, **d, a** des deux graphe (premier et dernier) sont des feuilles

Remarque : Un arbre est un graphe simple.

2.4) Théorème 1

Soit G = **(X, A) un graphe** sur **n = |X|**. Si **G** est connexe alors **|A| ≥ n – 1**

Démonstration :

Nous allons utiliser le principe de l'induction mathématique. L'idée est de montrer que cette inégalité est vraie pour **n = 1** (le cas de base) et de supposer qu'elle est vraie pour un certain n = k (hypothèse d'induction) pour prouver qu'elle est également vraie pour n = k + 1.

Cas de base (n = 1) : Lorsque n = 1, cela signifie que le graphe G **se compose d'un seul nœud**, et il n'y a **aucune arête** puisque le graphe est trivial. **Dans ce cas, |A| = 0**, et l'inégalité **|A| ≥ n - 1** est vérifiée **parce que 0 ≥1 - 1 est vrai.**

Hypothèse d'induction : Supposons maintenant que **l'inégalité |A| ≥ k - 1 soit vraie** pour un graphe **connexe G de taille n = k**, où **k est un nombre positif quelconque**.

Étape inductive : Nous allons maintenant montrer que l'inégalité est également vraie pour un graphe connexe G de taille **n = k + 1**. Supposons que nous avons un graphe **G' = (X', A')** avec **n = k + 1** nœuds.

Puisque G' est connexe, il existe un chemin entre deux nœuds quelconques. Supposons que nous avons un nœud **v** quelconque dans **G'**. Si nous retirons ce nœud **v** et toutes **les arêtes qui y sont incidentes**, nous obtenons un sous-graphe connexe **G''** de **G'** avec nœuds **X'' = X' - {v}** et arêtes **A'' = A' - {toutes les arêtes incidentes à v}.**

Par l'hypothèse d'induction, nous savons que **|A''| ≥ (k - 1).** De plus, lorsque nous rajoutons le nœud v et ses arêtes incidentes pour former G', nous ajoutons au moins une arête de plus à |A''|, **car G' est connexe et chaque nœud ajouté doit être relié à un nœud existant par au moins une arête.**

Donc, **|A'| = |A''| + au moins 1**, ce qui signifie **|A'| ≥ (k - 1) + 1 = k.**

Nous avons donc montré que si l'inégalité **|A| ≥ k - 1** est vraie pour un graphe connexe de **taille n = k**, alors elle est également vraie pour un graphe connexe de taille **n = k + 1.** Par conséquent, par le principe de l'induction mathématique, cette inégalité est vraie pour tous les graphes connexes de nœuds **n = k, k + 1, k + 2, etc.** En particulier, elle est vraie pour **n = 1**, **ce qui conclut la démonstration du théorème.**

2.5) Théorème 2

- Si dans **G** tous les sommets sont de degré **supérieur ou égal à 2**, alors **G possède** au moins un cycle.
- Si G est sans cycle, il possède au moins un sommet de degré 0 ou 1

Démonstration :

Puisque Tous les sommets du Graph sont de degré **supérieur ou égal à 2,** **A**ors Si on emprunte un chemin T , on peut toujours sortir de n'importe quel sommet visité (car ils sont de degré >=2) alors on est obligé à un certain instant de revisiter un sommet déjà visité **donc et on tombe sur un cycle**. Sinon on ne **s'arrête jamais → d'où pour que G soit sans cycle** il faut y avoir au moins un sommet de degré 0 ou 1

Corolaire :

- Un Arbre de n Sommet comporte (n-1) arêtes

Démonstration

La démonstration que tout arbre de n sommets comporte (n-1) arêtes est généralement faite par récurrence. Nous allons utiliser une preuve par induction mathématique pour établir ce résultat.

Cas de base (n=1) : Pour n=1, un arbre avec un seul sommet ne comporte aucune arête. Donc, (n-1)=0. L'affirmation est vraie pour n=1.

Hypothèse d'induction : Supposons que l'affirmation soit vraie pour un arbre de n sommets, c'est-à-dire qu'un arbre avec n sommets comporte (n-1) arêtes.

Étape inductive : Nous allons maintenant démontrer que l'affirmation est également vraie pour un arbre de (n+1) sommets.

Considérons un arbre **T avec (n+1)** sommets. Sélectionnons un sommet de cet arbre, appelons-le v, qui est une feuille, c'est-à-dire un sommet ayant un degré égal à 1. Supprimons ce sommet v de l'arbre T, ce qui laissera un arbre **T' avec n sommets**. Puisque v est une feuille, la suppression de v ne déconnecte pas les autres sommets de T', car ils sont toujours reliés par les arêtes restantes de T.

Par l'hypothèse d'induction, l'arbre **T**'avec n sommets comporte (n-1) arêtes. Lorsque nous réintégrons le sommet v, nous ajoutons une seule arête à T' pour le connecter au reste de l'arbre. Par conséquent, l'arbre T avec (n+1) sommets comporte (n-1) + 1 = n arêtes.

Nous avons donc montré que si l'affirmation est vraie pour un arbre de n sommets, alors elle est également vraie pour un arbre de (n+1) sommets. Puisque l'affirmation est vraie pour n=1 (cas de base), elle est vraie pour n=2, puis pour n=3, et ainsi de suite, par induction. Par conséquent, tout arbre de n sommets comporte effectivement (n-1) arêtes.

2.6) Théorème 3

- Si G est sans cycle, $|A| \leq n - 1$ (n nombre de sommets)

Démonstration

Pour démontrer le théorème qui affirme que si G est sans cycle, alors $|A| \leq n - 1$, nous allons utiliser une preuve par l'absurde. Cela signifie que nous allons supposer le contraire de ce que nous voulons prouver, puis montrer que cela conduit à une contradiction.

Supposons que G soit un graphe sans cycle (un graphe acyclique), mais que $|A| > n - 1$. Cela signifie qu'il y a plus d'arêtes que ce que l'on serait autorisé à avoir dans un graphe avec n sommets s'il était un arbre (un graphe connexe sans cycle).

Un graphe acyclique avec n sommets, s'il était un arbre, aurait au plus n - 1 arêtes. C'est un fait bien connu dans la théorie des graphes. Donc, nous pouvons écrire mathématiquement

$|A| \leq n-1$

Cependant, nous avons supposé que $|A| > n - 1$, ce qui est contraire à cette inégalité. Cela crée une contradiction, ce qui signifie que notre hypothèse selon laquelle G est un graphe sans cycle avec $|A| > n - 1$ est incorrecte.

Par conséquent, nous pouvons conclure que si G est un graphe sans cycle, alors |A| ne peut pas être strictement supérieur à n - 1. En d'autres termes, $|A| \leq n - 1$ lorsque G est un graphe sans cycle. Cela conclut la démonstration du théorème.

2.7) Propositions : Caractérisation des arbres

Soit G=(X, A) un graphe sur $n \geq 2$ Sommets. **G est un arbre alors** les propriétés suivantes sont équivalentes et caractérisent **G** :

- ✓ G comporte (n - 1) arêtes.
- ✓ G est connexe et sans cycle,
- ✓ G est connexe et **Minimal**,
    - Pour cette propriété (si on **supprime** une arête de G, il **n'est plus connexe**).
- ✓ G est sans cycle et **maximal**
    - Pour cette propriété (si on **ajoute** une arête a G, **on crée un cycle**)

2.8) Corollaire

Tout graphe connexe G = (X, A) possède un graphe partiel qui est un arbre.

2.9) Théorème 4

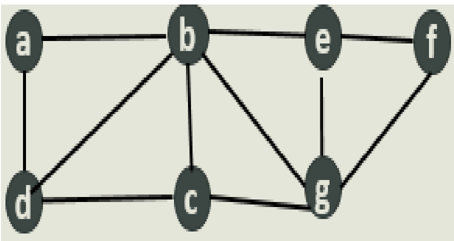
Un arbre G = (X, A) sur n ≥ 2 sommets admet au moins 2 sommets **pendants** (**sommets de degré 1**).

2.10) Théorème 5

Si G est un arbre alors tout couple de sommets de G est relie par une chaine élémentaire unique

5.2.1 Exercice 17 :

On considère le graphe non orienté suivant :



- Combien faut-il enlever d'arêtes à ce graphe pour le transformer en arbre ?
- Donnez un graphe partiel de ce graphe qui soit un arbre

Correction

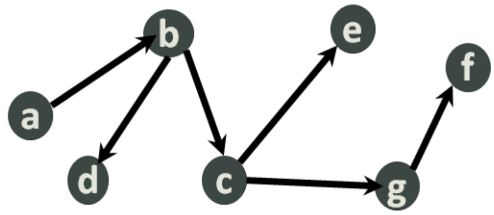
Le graphe comporte **7 sommets et 11 arêtes**. Pour le transformer en arbre il faudra donc enlever **5 arêtes**.

Par exemple, les arêtes **(f, g), (b, g), (b, c), (b, d) et (a, d).**

6 Arborescence

Une arborescence est **un graphe orienté sans circuit** admettant une **racine** $\boldsymbol{u \in S}$ telle que, pour tout autre sommet $\boldsymbol{v \in S}$, il existe un chemin unique allant de $\boldsymbol{u}$ vers $\boldsymbol{v}$ l'arborescence comporte $\boldsymbol{n}$ **sommets**, alors elle comporte exactement $\boldsymbol{n - 1}$ ***arcs***.

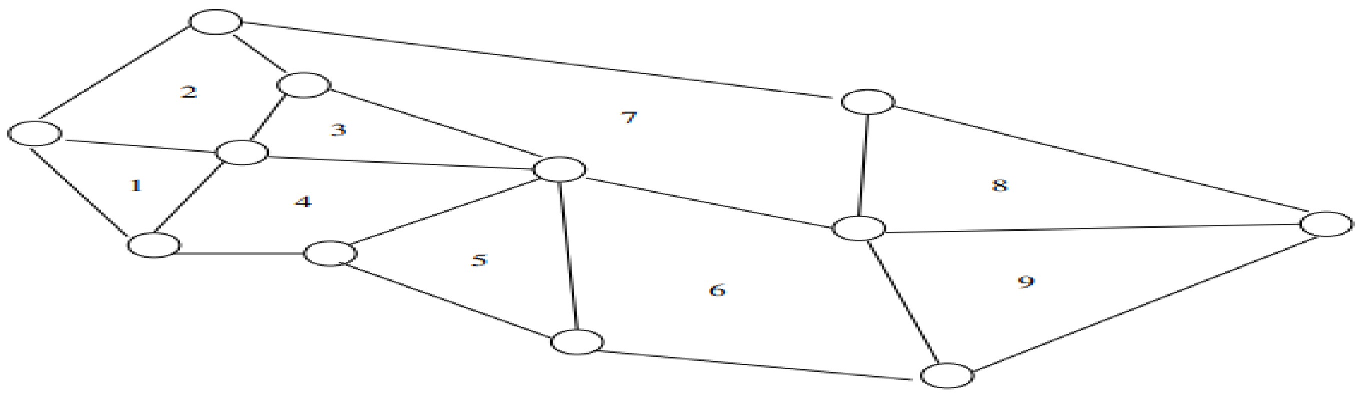
Par exemple, le graphe suivant est une arborescence de ***racine a***



Exercice 18 : L'ile du Nivéou, en Camargue, se consacre à la culture du riz. Sur cette ile se trouvent 9 champs entourés de murs et disposés de la façon suivante :

La culture du riz nécessite inonder périodiquement l'ensemble des champs. Cela est réalisé en ouvrant des vannes placées dans les murs séparant les champs et le Rhône ou les champs entre eux. Etant donné que l'installation d'une vanne est coûteuse, il s'agit de déterminer le nombre minimum de vannes et leur emplacement pour pouvoir, quand on le désire, inonder tous les champs.

Correction : Pour résoudre ce problème, on peut considérer le graphe non orienté comportant un sommet pour chaque intersection de mur, et une arête pour chaque mur :

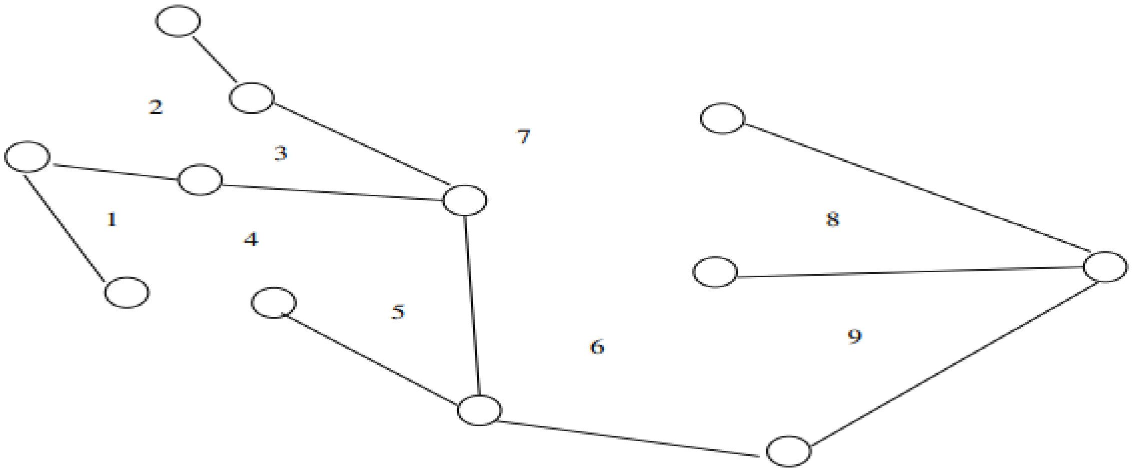


placer une vanne sur un mur➔ à supprimer l'arête correspondante dans le graphe, le problème revient à supprimer des arêtes jusqu'à ce que le graphe ne comporte plus de cycles

➔ Placer des vannes jusqu'à ce qu'il n'y ait plus de champ entouré de murs sans vanne). Comme on souhaite poser le moins de vannes possible, on cherche un graphe partiel sans cycle tel que si l'on rajoute une arête on crée un cycle :

- Selon les propositions **2 et 4** , il s'agit d'un arbre. Ici, étant donné que le graphe a 12 sommets et 20 arêtes, l'arbre devra posséder **12-1 = 11 arêtes** (**selon la proposition 1**),

→ et on devra donc installer **20-11=9** vannes. On obtiendra (par exemple) l'arbre suivant :



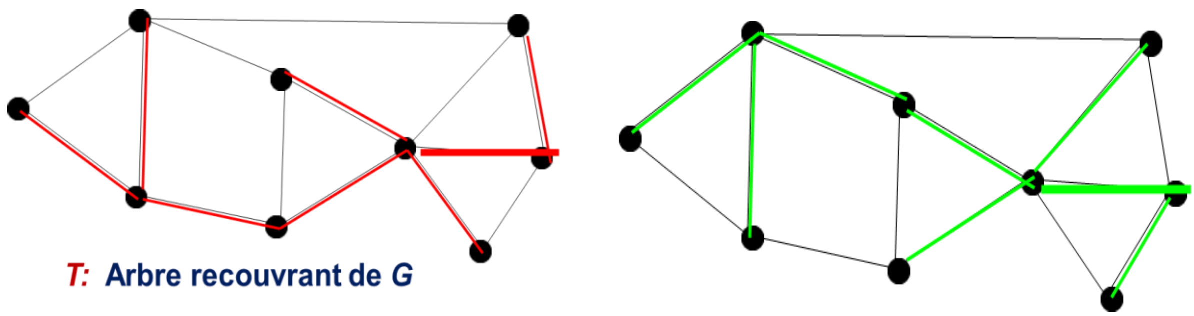
7 Arbre recouvrant

Un arbre recouvrant d'un Graphe **non orienté *G*** *est **un graphe partiel de G** formé par un arbre qui relie **tous les sommets de G**.*

Un Arbre recouvrant contient des sommets et des arêtes talque : :

- **sommets** : tous les sommets de *G*
- **arêtes** : certaines arêtes de *G*

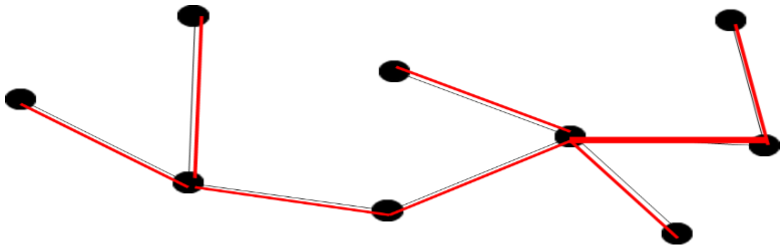
Exemple :



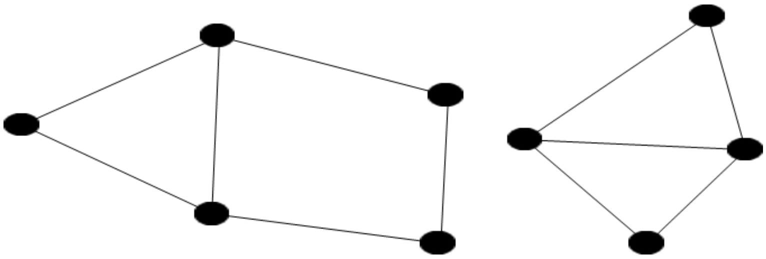
Dans l'exemple ci-dessus **les deux arbres construits** par des arrêtes en rouge et en vert sont des arbres recouvrant pour le même graphe **G** .

Remarques importantes :

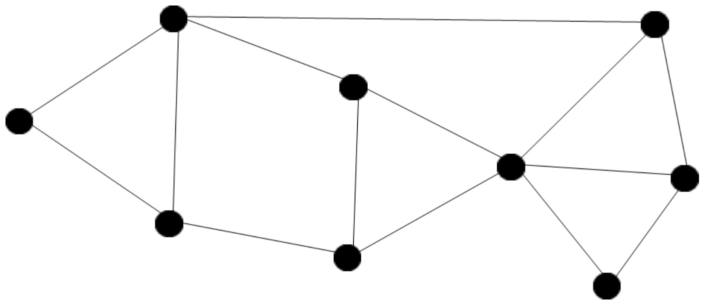
- Un graphe peut avoir **plusieurs** arbres recouvrant.
- Un arbre n'a qu'un seul arbre recouvrant, lui-même.



- Un graphe non connexe n'a aucun arbre recouvrant (Autrement dit, un graphe qui a un arbre recouvrant est forcément connexe.)

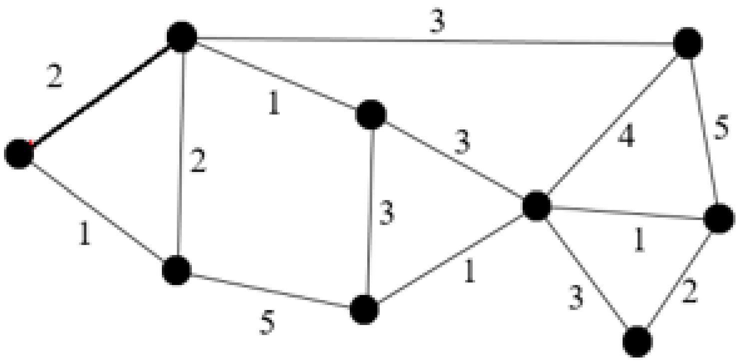


- **Un graphe connexe** a forcément (**au moins**) un arbre recouvrant.



8 Graphe valué

Un **graphe valué**, également appelé "**graphe pondéré**", est un type de graphe dans lequel chaque arête (ou lien) entre les nœuds (ou sommets) est associée à une valeur numérique, appelée "poids" ou "valeur". Ces poids représentent généralement des mesures de distance, de coût, de capacité, ou d'autres caractéristiques pertinentes pour le problème ou le domaine spécifique auquel le graphe est appliqué.

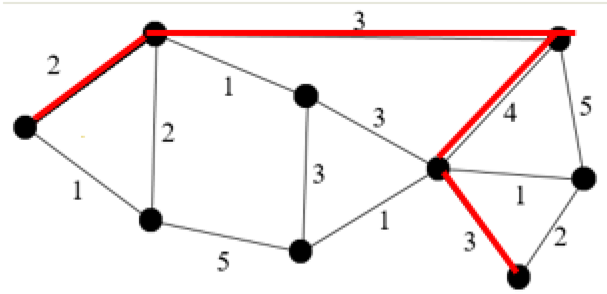


Les graphes valués sont couramment utilisés pour modéliser des situations dans lesquelles il est important de prendre en compte des facteurs quantitatifs pour résoudre un problème. Par exemple, un graphe valué peut être utilisé pour représenter un réseau de villes où les poids des arêtes représentent les distances entre les villes, ou pour modéliser un réseau de télécommunications où les poids indiquent la latence ou le coût de la transmission de données entre différents points.

Poids d'une chaine

Le poids d'une chaîne est la somme des poids des arêtes qui la composent

Dans cet exemple **la chaine en rouge a un poids égal à 12 (2 +. 3 + 4 + 3 =12)**

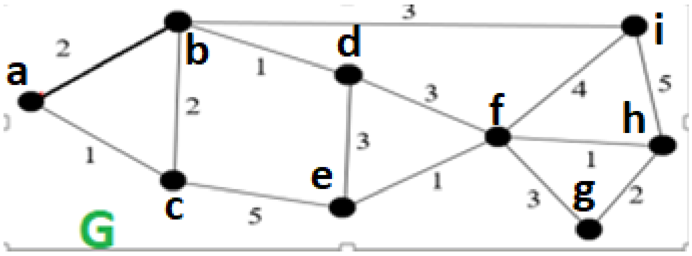


9 L'arbre recouvrant et Algorithmes de recherche d'un arbre couvrant de poids minimum

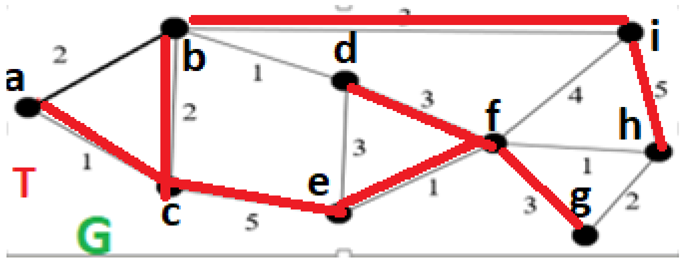
L'objectif principal de cette section est de découvrir comment, parmi tous les arbres recouvrants possibles d'un graphe, on peut identifier celui ayant le poids total le plus bas. Pour ce faire, nous explorerons des algorithmes clés, tels que l'algorithme de Kruskal et l'algorithme de Prim, qui nous permettent de résoudre ce problème de manière efficace. En comprenant ces méthodes, vous serez en mesure d'optimiser la conception et la gestion de réseaux, de minimiser les coûts, et d'atteindre des solutions efficaces pour de nombreux défis pratiques.

9.1 Calcul du poids d'un arbre couvrant

Étant donné un **graphe valué** connexe



- Une des arbres courant de Graphe **G est T en rouge defini comme suit :**
  - **Sommets : tous les sommets**
  - **Arêtes: {(a,c),(c,b),(b,i),(c,e),(e,f),(f,d),(f,g))**



T est donc an arbre couvrant de poids **23 (1 + 2 + 5 + 1 + 3 + 3 + 5 + 3 = 23)**

Question :

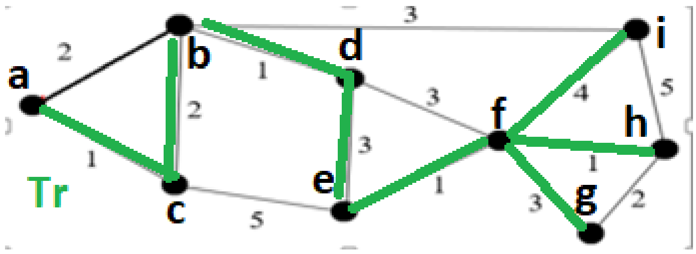
Peut-on faire mieux ? En trouvant un autre arbre couvrant ayant un poids < 23

Réponse :

En essayant on trouve le graphe **Tr** définit comme suit :

Sommets : tous les sommets bien surs

Arêtes: **{(a,c),(c,b),(b,d),(d,e),(e,f),(f,i),(f,g),(f,h))**



Tr est donc an arbre couvrant de poids 16 (**1 + 2 + 1 + 3 + 1 + 3 + 1 + 4 = 16)**

On remarque bien que le poids de l'arbre **Tr** est **inférieur** au poids de **T** mais la question persiste toujours (**Peut-on faire mieux ? :** un autre arbre couvrant ayant un poids <16)

Pour répondre définitivement à cette question on va avoir recours à des algorithmes qui permettent de construire un graphe partiel qui soit un arbre et pour lequel la somme des poids des arêtes est minimum tel que les deux algorithmes suivants qui seront l'objet de notre section suivantes :

- **Algorithme de Kruskal**
- **Algorithme de Prime**

9.2 Algorithme de Kruskal

L'algorithme de Kruskal est un puissant outil de la théorie des graphes, largement utilisé pour résoudre le problème de recherche de l'arbre recouvrant de poids minimum dans un graphe. Cet algorithme, du nom de son concepteur **Joseph Kruskal,** permet de trouver un sous-ensemble d'arêtes d'un graphe, formant un arbre recouvrant, dont la somme des poids est minimale.

Dans cette section, nous plongerons dans les détails de **l'algorithme de Kruskal** et explorerons sa logique fondamentale. Il repose sur une approche gloutonne, **sélectionnant progressivement les arêtes de poids minimum** tout **en évitant la formation de cycles** dans l'arbre recouvrant en cours de construction. L'algorithme est non seulement élégant par sa simplicité, mais il est également très efficace pour résoudre divers problèmes pratiques tels que la conception de réseaux de transport, la planification de câblage, ou la création d'infrastructures minimisant les coûts.

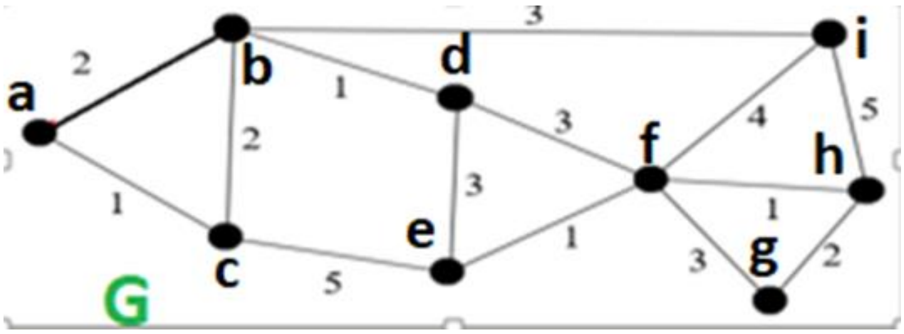
Comprendre l'algorithme de **Kruskal** et son fonctionnement vous donnera un outil précieux pour résoudre des problèmes d'optimisation dans de nombreux domaines, où la minimisation des coûts ou des distances joue un rôle essentiel. Nous allons examiner étape par étape comment cet algorithme parvient à déterminer un arbre recouvrant du poids minimum, ouvrant ainsi la voie à des solutions optimales dans des situations réelles.

1- **Les Etapes de l'algorithme de Kruskal**

- ❑ Initialiser ***T*** **(T est l'arbre couvrant qu'on cherche)**:
  - **sommets** : **tous les sommets de *G*** et **arêtes** : **aucune**
- ❑ Soit A={*e1, e2 … en*} l'ensemble arêtes de **G** triées par poids croissant
- ❑ Pour ***ei*** de ***A*** faire :
  - ❑Si ***ei*** permet de connecter deux composantes connexes de ***T*** **sans cycle**,
    - ❑ alors ajouter **ei** à ***T***
  - ❑**sinon** ne rien faire
- ❑ Fin Pour : S'arrêter (quand il n'y a plus d'arêtes)
- ❑ Retourner ***T***

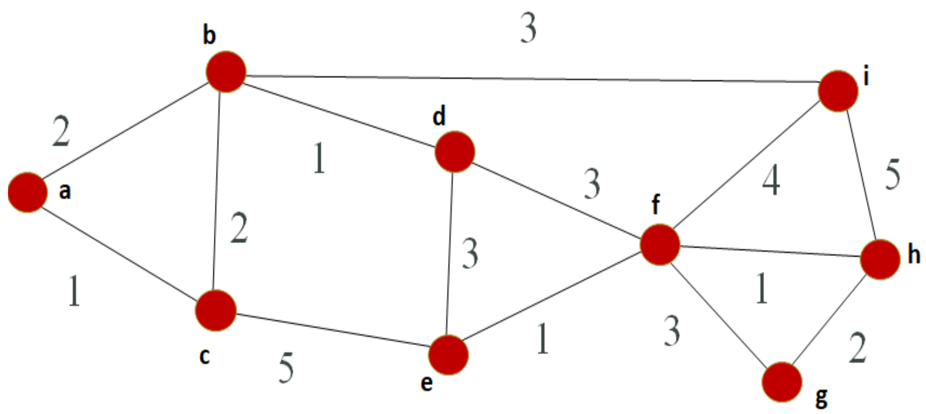
2- **Illustration de l'algorithme de Kruskal sur un graphe**

Considérant le même **graphe valué G** vu dans la section précédente dont nous rappelons dabs la figure ci-dessous :



- **Etape 1** : **Initialisation →** sélectionner tous les sommets : La sélection est faite par **la coloration des sommets en** couleur **rouge** Donc l'arbre couvrant sera

TC= (S , { }) avec **S = tous les sommets**

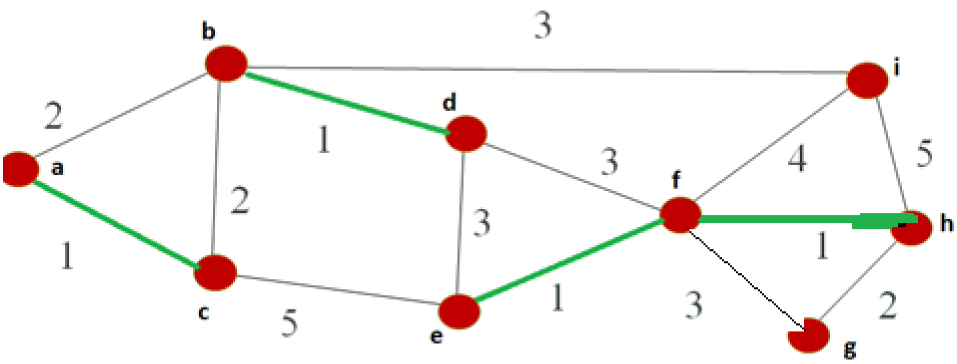


Etape 2 :

Etape 2. 1 Choisir les arêtes de poids minimum en évitant le cycle

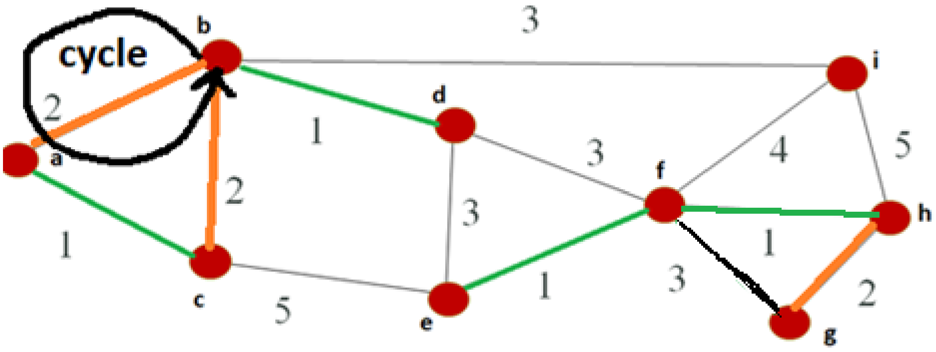
- Dans cette étape les arêtes choisis sont de poids 1 et sont {(a,c),(b,d),(e,f),(f,g)} et puisque aucune de ces arêtes ne provoque un cycle on les garde donc L'arbre couvrant TC (S,A) sera alimentée par ces arrêtes devient :

  - **TC (S , {(a,c),(b,d),(e,f),(f,h)})**



Etape 2. 2 : Choisir les arêtes de poids minimum en évitant le cycle

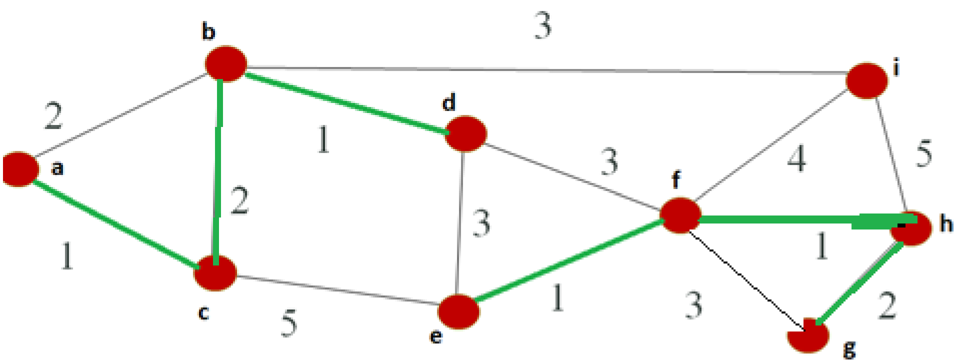
- Dans cette étape **les arêtes choisis sont de poids 2 → {(a,b),(b,c),(h,g)}**



Puisque l'arête **(a,b) provoquera un cycle il sera éliminé des arrêtes et** on grade dans cette étape seulement **{(b,c),(h,g)} et l'arbre TC devient :**

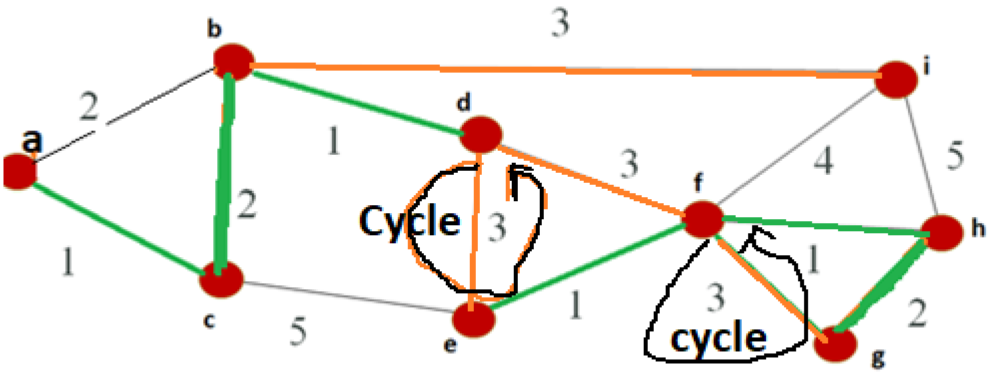
- **TC (S , {(a,c),(b,d),(e,f),(f,h), (b,c),(h,g)})**

-



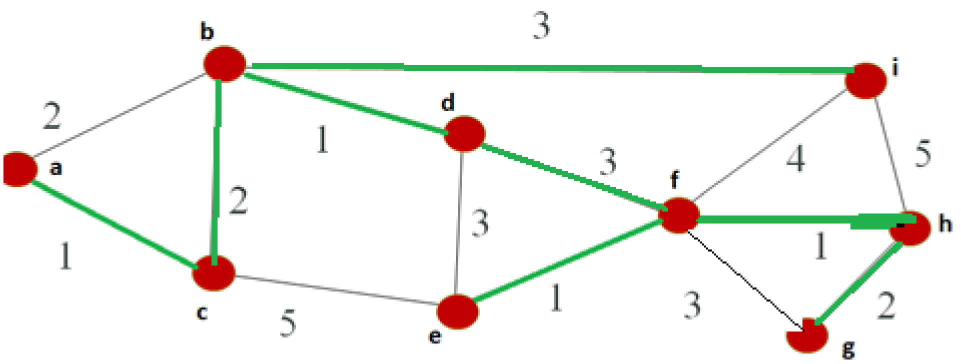
Etape 2. 3 : Choisir les arêtes de poids minimum en évitant le cycle

- Dans cette étape les arêtes choisis sont de **poids 3→ {(b,i),(d,f),(d,e),(f,g)}**



Puisque les deux **arêtes (d,e) et (fng) provoqueront un cycle il seront éliminées des arrêtes et** on grade dans cette étape seulement **{(b,i),(d,f)} et l'arbre TC devient :**

- **TC (S , {(a,c),(b,d),(e,f),(f,h), (b,c),(h,g), (b,i),(d,f))**



Etape 2. 4 Choisir les arêtes de poids minimum en évitant le cycle

- Dans cette étape les arêtes choisis sont de **poids 4→ {(f,i)}**

Puisque cette arête provoquera un cycle il sera éliminé dy graphe L'arbre TC reste la même que dans l'étape précédente : ie TC (S , {(a,c),(b,d),(e,f),(f,h), (b,c),(h,g), (b,i),(d,f))

Etape 2. 5 Choisir les arêtes de poids minimum en évitant le cycle

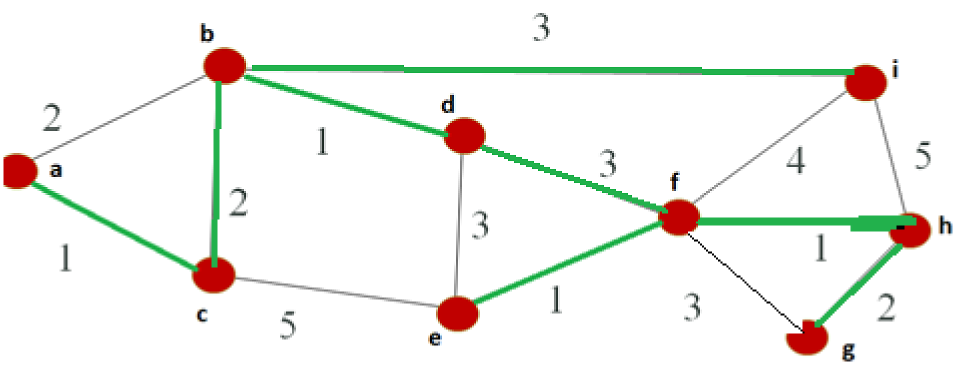
Dans cette étape les arêtes choisis sont de **poids 5→ {(h,i)}**

Puisque cette arête provoquera un cycle il sera éliminé dy graphe L'arbre TC reste la même que dans l'étape

précédente: ie TC (S , {(a,c),(b,d),(e,f),(f,h), (b,c),(h,g), (b,i),(d,f))

Etape 2. 5 : il ne reste aucune arrête alors Fin de L'algorithme et

Le Result final → TC (S , {(a,c),(b,d),(e,f),(f,h), (b,c),(h,g), (b,i),(d,f))

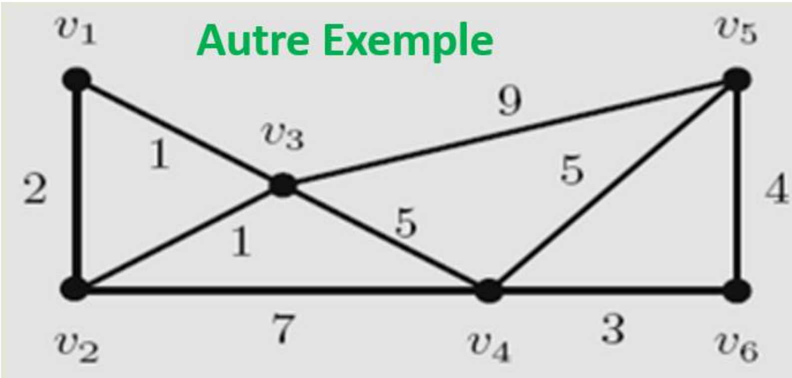


Avec un poids de 14 (1 + 1 + 1 + 1 + 2 + 2 + 3 + 3 = 14)

Peut-on faire mieux ? Évidemment non

a) Autre Illustration de l'algorithme de Kruskal sur un graphe en utilisant un tableau

Soit le graphe suivant :



L'ensemble des arrêtes du graphe sont :

$$e_1 = \{v_1, v_3\}, e_2 = \{v_2, v_3\}, e_3 = \{v_1, v_2\},$$
$$e_4 = \{v_4, v_6\}, e_5 = \{v_5, v_6\}, e_6 = \{v_4, v_5\},$$
$$e_7 = \{v_3, v_4\}, e_8 = \{v_2, v_4\}, e_9 = \{v_3, v_5\}.$$

En appliquant l'algorithme de Kruskal par intégration on aura avec T est l'arbre couvrant a poids minimal

Itér (i)	e_i	$W(e_i)$	$\in T$
1	$\{v_1 , v_3\}$	1	oui
2	$\{v_2 , v_3\}$	1	oui
3	$\{v_1 , v_2\}$	2	non
4	$\{v_4 , v_6\}$	3	oui
5	$\{v_5 , v_6\}$	4	oui
6	$\{v_4 , v_5\}$	5	non
7	$\{v_3 , v_4\}$	5	oui

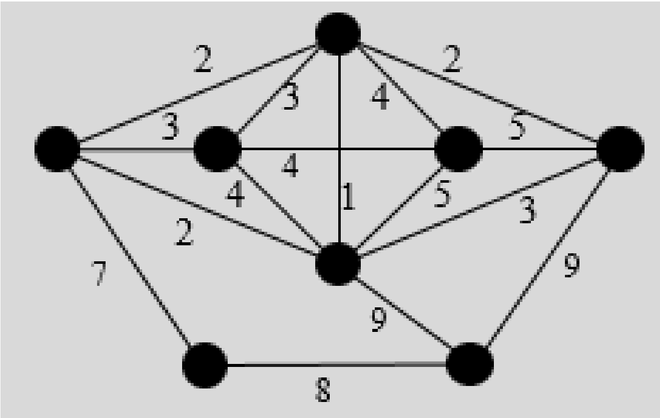
T est donc formé de

- Sommets : tous les sommets
- Arêtes : {(v1,v2) ,(v2,v3) ,(v4,v6) ,(v(,v6),(v3,v4)}

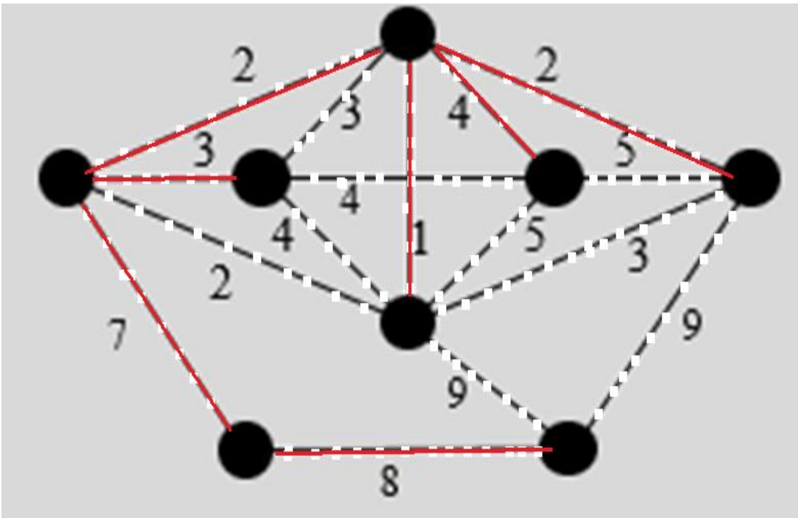
Et son poids est de : 14

5.2.2 Exercice 19

Appliquer l'algorithme de **KRUSKAL** au graphe suivant.

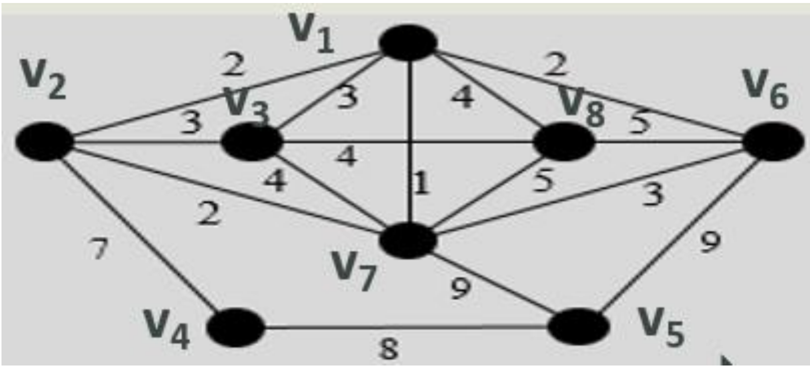


Solution



Poids est → 1 + 2 + 2 + 3 + 4 + 7 + 8 = 27

Solution : tableau des itérations



- **Trie des arrêtes par ordre croissant**

{v_1 , v_7} {v_2 , v_6} {v_1 , v_2} {v_7 , v_2} {v_2 , v_3}
{v_3 , v_1} {v_7 , v_6} {v_1 , v_8} {v_3 , v_7} {v_7 , v_8}
{v_8 , v_6} {v_2 , v_4} {v_4 , v_5} {v_6 , v_5} {v_7 , v_5}

Calcul des arêtes participant à l'arbre par tableau d'itération

Itér (i)	e_i	$W(e_i)$	$\in T$
1	$\{v_1, v_7\}$	1	oui
2	$\{v_2, v_6\}$	2	oui
3	$\{v_1, v_2\}$	2	oui
4	$\{v_7, v_2\}$	2	non
5	$\{v_2, v_3\}$	3	oui
6	$\{v_3, v_1\}$	3	non
7	$\{v_7, v_6\}$	3	non
8	$\{v_1, v_8\}$	4	oui
9	$\{v_3, v_7\}$	4	non
10	$\{v_7, v_8\}$	5	non
11	$\{v_8, v_6\}$	5	non
12	$\{v_2, v_4\}$	7	oui
13	$\{v_4, v_5\}$	8	oui
14	$\{v_6, v_5\}$	9	non
15	$\{v_7, v_5\}$	9	non

Poids est 1 + 2 + 2 + 3 + 4 + 7 + 8 = **27**

6.2) Algorithme de Prime

L'algorithme de prime est une méthode essentielle en théorie des graphes et en optimisation combinatoire, principalement utilisée pour trouver un arbre couvrant de poids minimum dans un graphe pondéré. Conçu par le mathématicien tchèque **Vojtěch Jarník** en 1930 et ultérieurement popularisé par le mathématicien américain **Robert C. Prim** en **1957**, cet algorithme offre une solution efficace pour résoudre des problèmes tels que le problème du voyageur de commerce et la conception de réseaux de communication efficaces. L'objectif fondamental de l'algorithme de prime est de sélectionner progressivement les arêtes du graphe de manière à construire un arbre couvrant qui relie tous les sommets, en minimisant la somme des poids des arêtes. Dans ce contexte, nous explorerons les principes de base de l'algorithme de prime.

a. Les Etapes de l'algorithme de Prime

- ❑ Initialiser **T** avec (**T** est l'arbre couvrant qu'on cherche)
- **Sommets** : **un sommet** de G (ce sommet est choisi au hasard
- **Arêtes** : aucune
- ❑ Répéter :

- ❑ Trouver toutes **les arêtes de G** qui relient un sommet de T et un sommet **extérieur à T**
- ❑ Parmi celles-ci, choisir une arête **de poids le plus petit possible**
- ❑ Ajouter à **T** cette arête et le sommet correspondant

❑ **S'arrêter** dès que tous les sommets de **G sont dans T**

❑ **Retourner T**

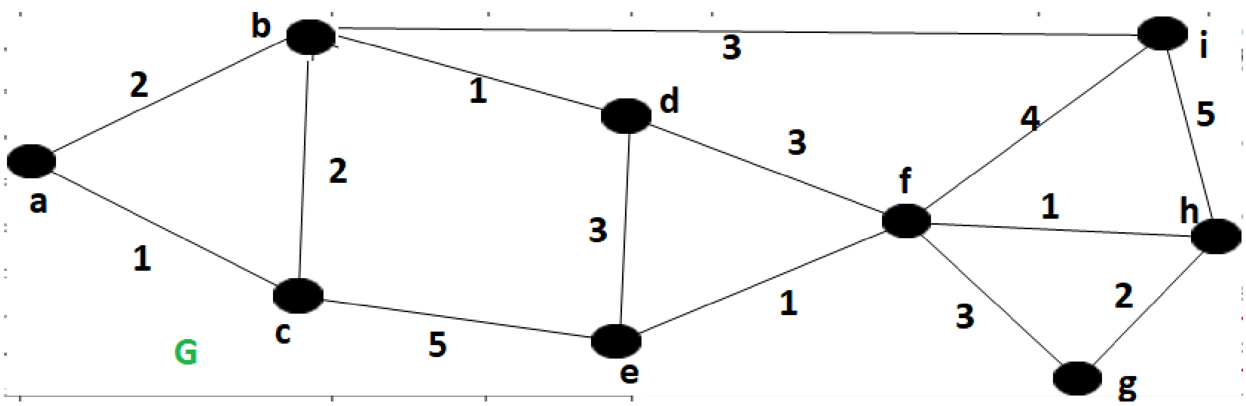
b. Illustration de l'algorithme de Prime sur un graphe

On peut illustrer l'application de **l'algorithme de prime** par deux méthodes

- A l'aide des dessins en parcourant le graphe et dessinant à chaque itération le graphe sur lequel on mentionne graphiquement les **sommets et les arêtes** progressivement à l'aide de dessin
- A l'aide **d'un tableau** de l'itération sur lequel on mentionne les sommets et les arêtes, cette méthode est la plus pratique

b.1) Illustration de l'algorithme de Prime à l'aide des dessins

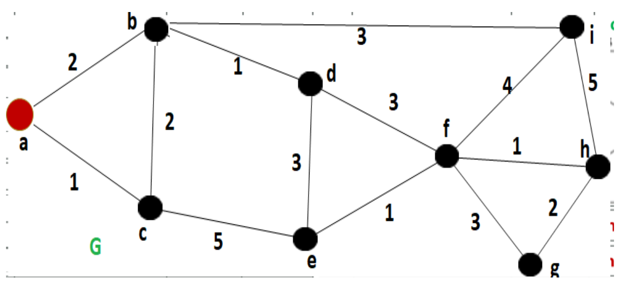
Considérons le même graphe G adopté dans la section précédant



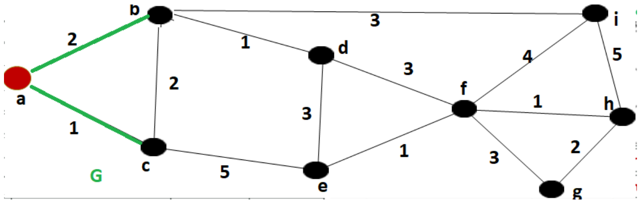
Etape 1 : Initialisation de l'arbre couvrant de poids minimal **TP** :

- **Sommets** : **a** (choisi au hasard marqué en **rouge**)
- **Arêtes :** **aucun**

Fin Etape 1

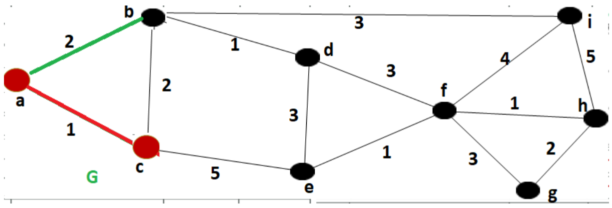


Etape 2 :



Itération 1 :

<Tous les sommets sont dans TP ?>→ NON

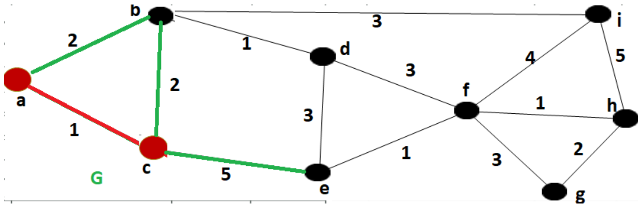


- Marquer **toutes les arêtes** incidentes au sommet a (en **vert (a,b) et (a,c)**)
- Puis choisir l'arête de plus de petit poids parmi les arêtes marquées et l'ajouter à l'arbre TP

TP devient : Sommets (a et c)

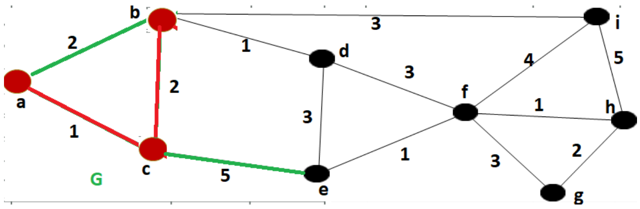
- **Arêtes : (a,c)**

Itération 2 :



<Tous les sommets sont dans TP ?>→ Non

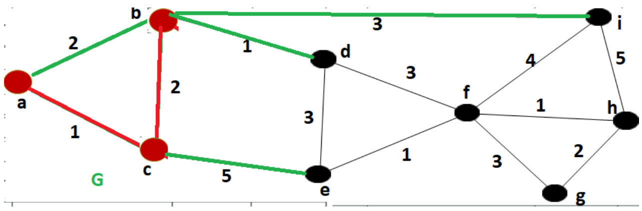
- Marquer toutes **les arêtes** incidentes au sommet **c** (en **vert (c,b) et (a,e)**)
- Puis choisir l'arête de plus de petit poids parmi les arêtes marquées et l'ajouter à l'arbre TP



TP devient : Sommets (a et c et b)

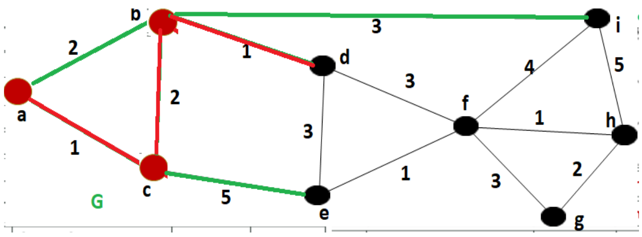
Arêtes : { (a,c), (b,c) }

Itération 3 :



<Tous les sommets sont dans TP ?> → Non

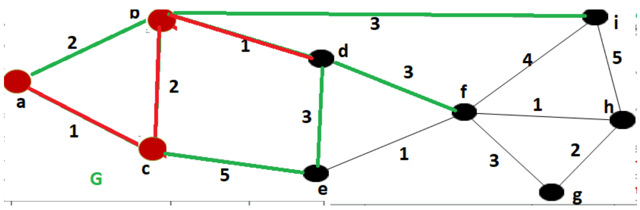
- Marquer toutes **les arêtes** incidentes au sommet **b** (**en vert (b,i) et (b,d)**)
- Puis choisir l'arête de plus de petit poids parmi toutes les arêtes candidates marquées en verts et l'ajouter à l'arbre TP



TP devient : Sommets (a et c et b et d)

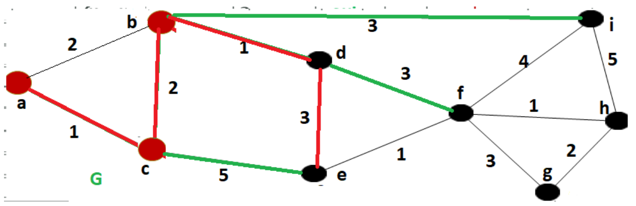
Arêtes : { (a,c), (b,c) ,(b,d)}

Itération 4 :



<Tous les sommets sont dans TP ? >→ Non

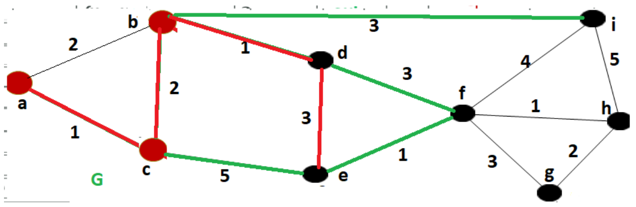
- Marquer toutes **les arêtes** incidentes au sommet **d** (en vert (d,f) et (d,e))
- Puis choisir l'arête de plus de petit poids parmi toutes les arêtes candidates marquées en verts et l'ajouter à l'arbre TP qui est **(a,b)**
- **(a,b)** provoque un cycle il faut l'éliminer et passer à l'autre qui est **(d,e)** ou **(d,i)** **(prenons (d,e))**



TP devient : Sommets (a et c et b et d et e)

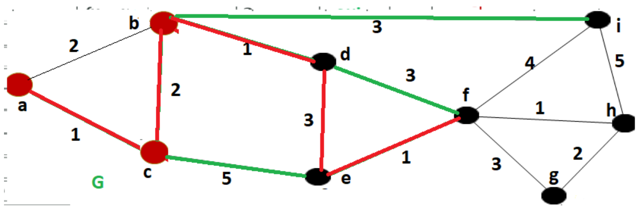
Arêtes : { (a,c), (b,c) ,(b,d), (d,e) }

Itération 4 :



<Tous les sommets sont dans TP ? →> Non

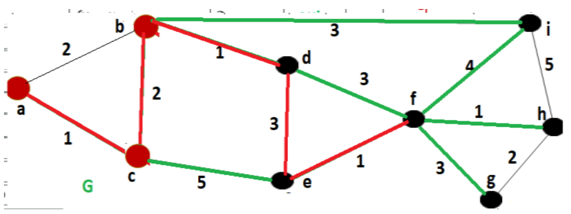
- Marquer toutes **les arêtes** incidentes au sommet **e** (en vert (e,f))
- Puis choisir l'arête de plus de petit poids parmi toutes les arêtes candidates marquées en verts et l'ajouter à l'arbre TP qui est **(e,f)**



TP devient : Sommets (a et c et b et d et e et f)

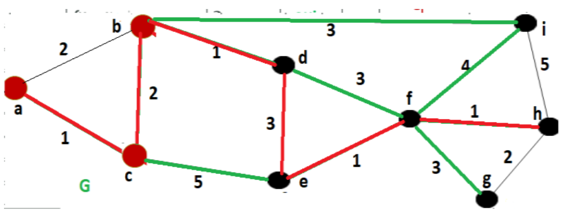
Arêtes : { (a,c), (b,c) ,(b,d), (d,e) ,(e,f)}

Itération 5 :



<Tous les sommets sont dans TP ? >→ Non

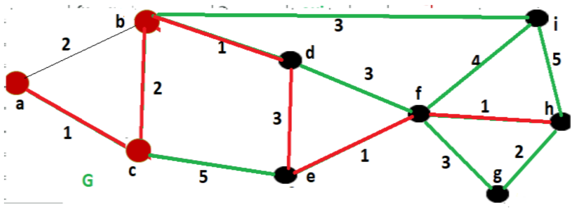
- Marquer toutes **les arêtes** incidentes au sommet **f** (en vert (f,i), (f,h), (f,g))
- Puis choisir l'arête **de plus de petit poids** parmi toutes les arêtes candidates marquées **en verts** et l'ajouter à l'arbre TP qui est **(f ,h)**



TP devient : Sommets (a et c et b et d et e et f)

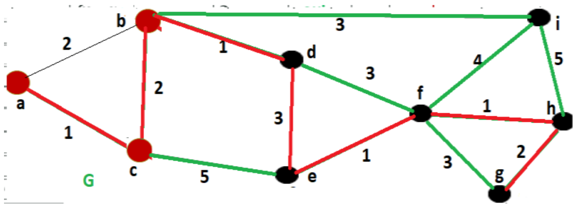
Arêtes : { (a,c), (b,c) ,(b,d), (d,e) ,(e,f)(f,h)}

Itération 6 :



<Tous les sommets sont dans TP ?> → Non

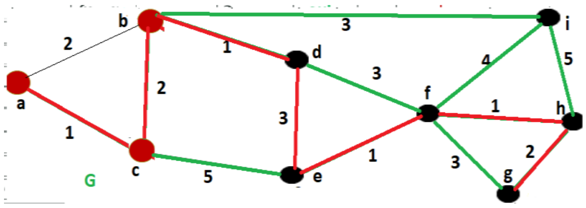
- Marquer toutes **les arêtes** incidentes au sommet **h** (**en vert (h,i), (h,g)**)
- Puis choisir l'arête **de plus de petit poids** parmi toutes les arêtes candidates marquées **en verts** et l'ajouter à l'arbre TP qui est **(h ,g)**



TP devient : Sommets (a et c et b et d et e et f)

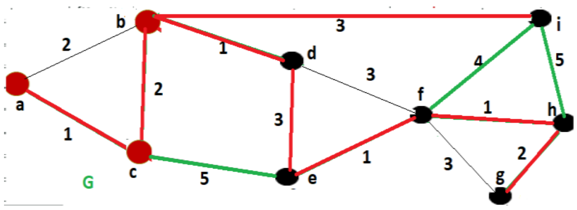
Arêtes : { (a,c), (b,c) ,(b,d), (d,e) ,(e,f)(f,h),(h,g)}

Itération 7 :



<Tous les sommets sont dans TP ? >→ Non

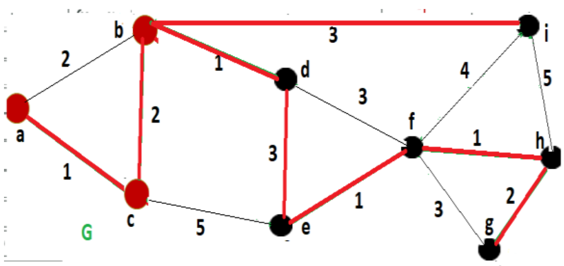
- Marquer toutes **les arêtes** incidentes au sommet **g** (**déjà fait dans les itération précédentes)**
- Puis choisir l'arête **de plus de petit poids** parmi toutes les arêtes candidates marquées **en verts** et l'ajouter à l'arbre TP
  - **Les arêtes (f,g) et (d,f) provoqueront un cycle, il faut les éliminer et garder (b,i)**



TP devient : Sommets (a et c et b et d et e et f et i)

Arêtes : { (a,c), (b,c) ,(b,d), (d,e) ,(e,f)(f,h),(h,g),(b,i)}

Itération 7 :



<Tous les sommets sont dans TP ? >→ OUI

→ Fin des itérations

➔ Fin du programme le résultat est :

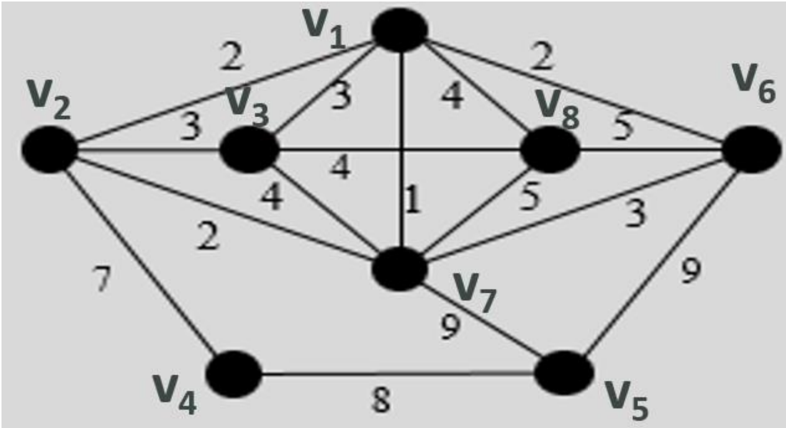
TP :

- **Sommets (a et c et b et d et e et f et i)**
- **Arêtes : { (a,c), (b,c) ,(b,d), (d,e) ,(e,f)(f,h),(h,g),(b,i)}**

Le poids de cet arbre est de **1 + 2 + 1 + 3 + 3 + 1 + 1 +2 = 14** → et c'est même résultat trouvé en appliquant l'algorithme de **Kruskal**

5.2.3 *Exercice 20*

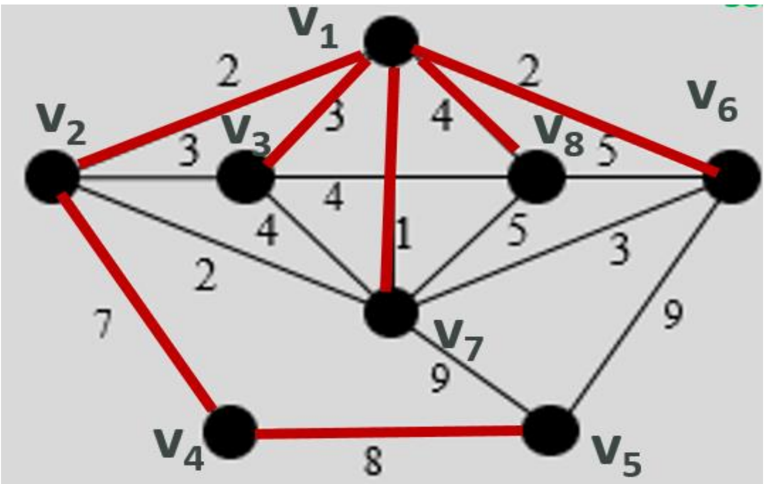
Appliquer l'algorithme de **Prime** au graphe suivant en commençant par un sommet de votre choix, **écrire l'arbre recouvrant de poids résultat de l'algorithme**



- **Solution finale** après application de l'algorithme de krsuskal **:**

L'arbre T = ((V1,V2), (v1,V3) (V2,v4) (v4,v5), (V1,V6) (V1,V7) (V1,V8))

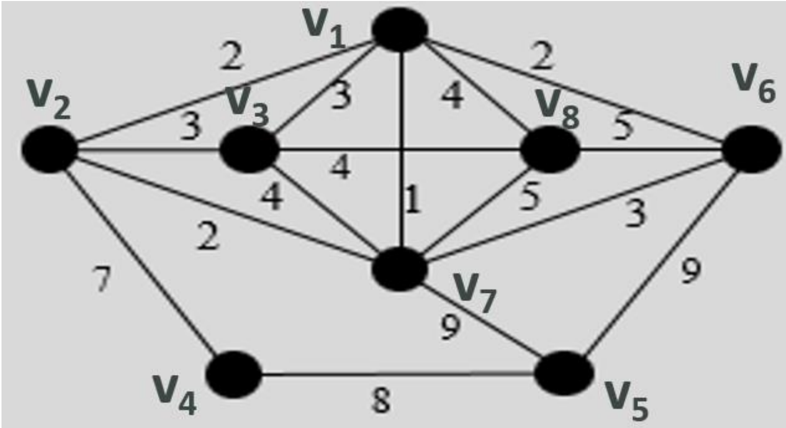
Poids : 27



b.2) Illustration de l'algorithme de Prime à l'aide de tableau des itérations

On peut appliquer l'algorithme de prime à l'aide d'un tableau de l'itération comme suit :

Considérons le même graphe étudié dans l'exercice 16 précèdent et cherchons son arbre couvrant de poids minium



Traçant un tableau qui contient **dans sa première ligne** tous les sommets, chacune dans une colonne

- **La ligne initialisation** on donne un cout de **+ l'infini** (+∞**)** à chaque arrêt possible
- **Dans les lignes itérations**
  - **Itér 1 :** on choisit un sommet au hasard ici **V1**
    - Les cases correspondants à chaque sommet de la colonne liant **V1 avec les sommets des colonnes de la ligne (sommets) et qui forment des** arêtes du graphe sont remplis par le « **poids/sommet »** (**exp** : pour la colonne V2 on écrit 2/V1 , pour V8 on ecrit 4/V1) , les autres cases qui ne froment pas d'arêtes sont laissés vides
    - **Itération suivante** la case (**iter/sommets**) est affecté par le sommet de l'itération précédente qui propose le **plus petit poids**) (**expl** dans l'itér2/Sommets on a **V7, car il propose un poids minimal (1) dans l'itération précédente)**
    - **Même procédé pour les itérations qui suivent jusqu'à ce qu'on visite tous les sommets.**
    - **Le résultat final est récupéré à partir du tableau en liant la colonne avec la dernière ligne mentionnée, ainsi on aura**
- **La colonne V1** : n'a aucun sommet d'intersection dans les ligne
- **La colonne V2** : on a le sommet V1 avec le poids 2 qui est le dernier reporté →(V2,V1)
- **La colonne V3** : on a le sommet V1 avec le poids 3 qui est le dernier reporté→ (V3,V1)
- **La colonne V4** : on a le sommet V2 avec le poids 7 qui est le dernier reporté→ (V4,V2)
- **La colonne V5** : on a le sommet V4 avec le poids 8 qui est le dernier reporté→ (V5,V4)
- **La colonne V6** : on a le sommet V1 avec le poids 2 qui est le dernier reporté→ (V6,V1)
- **La colonne V7** : on a le sommet V1 avec le poids 1 qui est le dernier reporté→ (V7,V1)
- **La colonne V8 :** on a le sommet V1 avec le poids 4 qui est le dernier reporté→ (V8,V1)

L'arbre ainsi obtenu est : TP :

Sommets : tous les sommets

Arêtes : {(V2,V1), (V3,V1), (V4,V2), (V5,V4), (V6,V1), (V7,V1), (V8,V1)}

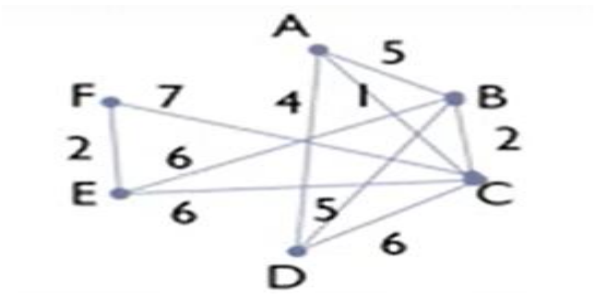
Poids = **2 + 3 + 7 + 8 + 2 + 1 + 4 =27**

	Sommets	V1	V2	V3	V4	V5	V6	V7	V8
Itération	**Initialisation**	∞ /	∞ /	∞ /	∞ /	∞ /	∞ /	∞ /	∞ /
Itér 1	**V1**		2/V1	3/V1			2/V1	1/V1	4/V1
Itér 2	**V7**		2/V1	3/V1		9/V7	2/V1		4/V1
Itér 3	**V2**			3/V1	7/V2	9/V7	2/V1		4/V1
Itér 4	**V6**			3/V1	7/V2	9/V7			4/V1
Itér 5	**V3**				7/V2	9/V7			4/V1
Itér 6	**V8**				7/V2	9/V7			
Itér 7	**V4**					8/V4			
Itér 8	**V5**								
	Total		2	3	7	8	2	1	4

5.2.4 Exercice 21

Appliquer l'algorithme de **Prime** au graphe suivant en utilisant la **technique de tableau d'itérations.**

Afficher l'arbre recouvrant de poids minimal résultant



Solution : Tableau des itérations

sommet	A	B	C	D	E	F
init	∞ /	∞ /	∞ /	∞ /	∞ /	∞ /
A		5 / A	1/ A	4 /A		
C		2 / C		4 / A	6 / C	7 /C
B				4 / A	6 / C	7 / C
D					6 / C	7 / C
E						2 / E
F						

L'Arbre recouvrant de poids minimal est: T = { (A,C), (B,C), (A,D),(C,E),(E,F)}

Le poids de cet arbre est 15 comme suit ; 1 + 2 + 4 + 6 + 2 =15

Partie 2) Théorie de jeux

Bienvenue à cette introduction à la théorie des jeux, un domaine fascinant et puissant de la science économique, des mathématiques et de la stratégie. La théorie des jeux offre un cadre analytique pour étudier les interactions entre les individus, les entreprises et même les nations. À travers cette exploration, nous allons plonger dans le monde des décisions stratégiques, des conflits d'intérêts, et des jeux d'optimisation. Cette partie de cours vous fournira les bases nécessaires pour comprendre les concepts clés de la théorie des jeux, ainsi que ses applications dans divers domaines tels que l'économie, la politique, la biologie et la psychologie. Préparez-vous à explorer un univers passionnant où les choix, les réponses rationnelles, et les conséquences se combinent pour révéler des insights profonds sur la nature humaine et les mécanismes de prise de décision. Dans cette partie de cours nous allons aborder les concepts suivant :

- ✓ Définition de jeux, forme normale et extensive
- ✓ Les jeux simultanés (statiques informations imparfaites)
- ✓ Les jeux séquentiels (dynamiques information parfaites)
- ✓ Résolution et équilibre du jeu :
    - Équilibre de Nash,
    - Le dilemme du prisonnier,
    - Stratégie dominante,
    - Jeu de la poule mouillée,
- Fonctions de meilleures réponses,
- Jeu en stratégies pures et jeu en stratégies mixtes

Chapitre 6) Définition de jeux, forme normale et extensive

Pour comprendre en profondeur les concepts clés de cette théorie, il est essentiel de débuter par une solide compréhension des définitions et des notations qui sous-tendent cette discipline.

Nous commencerons par étudier la forme normale d'un jeu, un modèle de représentation classique qui décrit les interactions stratégiques entre les joueurs sous forme de matrice. Cette représentation permet d'analyser les décisions des joueurs et de déterminer les résultats possibles du jeu.

Ensuite, nous explorerons la forme extensive, une représentation qui met en avant les jeux séquentiels. Dans un jeu séquentiel, les joueurs prennent leurs décisions tour à tour, ce qui introduit une dimension temporelle cruciale. Nous examinerons comment les arbres de décision sont utilisés pour modéliser ces jeux et comment ils permettent d'analyser les stratégies optimales des joueurs.

Enfin, nous aborderons les jeux simultanés, où les joueurs prennent leurs décisions en même temps, sans connaître les choix des autres. Nous verrons comment les équilibres de Nash sont utilisés pour analyser ces situations de jeu, où chaque joueur cherche à maximiser son gain tout en tenant compte des actions des autres.

Ce chapitre introductif jettera les bases nécessaires pour explorer plus en profondeur les concepts, les modèles **et les applications de la théorie des jeux tout au long de ce livre.**

1 Qu'est-ce qu'un jeu ?

La théorie des jeux est une branche des mathématiques et de l'économie qui étudie les interactions stratégiques entre différents acteurs, appelés joueurs, pour comprendre leurs décisions et les résultats possibles. Un jeu, dans ce contexte, peut être défini comme une situation où plusieurs acteurs font des choix stratégiques qui affectent mutuellement leurs gains ou pertes, et où les résultats dépendent des décisions de tous les joueurs.

- **Un jeu se compose donc de :**
  - **Un ensemble de joueurs.**
  - **Un ensemble de stratégies/ Actions** pour chaque joueur.
  - **Des gains/ Play-off** associés à chaque stratégie des joueurs.

1.1 Exemple introductif

Voilà un exemple très simple de jeu entre 2 agents (sous forme normale)

		Joueur B	
		L	R
Joueur A	U	(3,9)	(1,8)
	D	(0,0)	(2,1)

C’est un Jeu à 2 joueurs avec 2 stratégies possibles chacun.

- Les joueurs s’appellent **A** et **B**.
- Le joueur **A** a deux stratégies : “**up**” **(U)** ou “**down**” **(D).**
- Le joueur **B** a deux stratégies : “**Left**” **(L)** ou “**Right**” **(R).**
- La matrice des gains est représentée (**(sous forme normale)** comme suit :

Les gains du joueur **A** sont en **rouge** :

- 3 ou 0 si B joue “Left” (L)
- 1 ou 2 Si B joue Rigth (R)

Les gains du joueur **B** sont en **noir** :

- 9 ou 8 si A joue “UP” (U)
- 0 ou 1 Si A joue “Down” (D)

Exemples :

- Si **A** joue **Up** et **B** joue **Right** alors **A** gagne **1** et **B** gagne **8**
- Si **A** joue **Down** et **B** joue **Right** alors **A** gagne **2** et **B** gagne **1**
- Si **A** joue **Down** et **B** joue **Right** alors **A** gagne **2** et **B** gagne **1**

1.2 Résultat d’un jeu

Une situation (**Résultat possible**) de jeu est une paire (ex : (**U**,**R**)) où le premier élément (**U**) est la stratégie choisie par le joueur **A** et le deuxième élément (**R**) est la stratégie choisie par le joueur **B**

Quelle est donc le résultat de ce jeu ?

		Joueur B	
		L	R
Joueur A	U	(3,9)	(1,8)
	D	(0,0)	(2,1)

- **(U,R)** est-il un **résultat** possible ? Vérifions :
  - Si **B** joue R **(right)**alors la meilleure réponse de A est **D (down)**. Ainsi les gains de **A** passeront de 1 à 2. → Donc **(U, R)** n’est pas possible.
- **(D,R)** est-il un **résultat** possible ? Vérifions :
  - Si **B** joue **Right** alors la meilleure réponse de A est **Down**.
  - Si **A** joue **Down** alors la meilleure réponse de B est **Right**. → Donc, (**D,R**) **est possible.**

- **(D,L)** est-il un résultat possible ? Vérifions :
    - Si **A** joue **D(down)**, la meilleure réponse de **B** est **R**, → donc (**D,L**) n'est **pas possible**.
- **(U,L)** est-il un résultat possible ? Vérifions :
    - Si **A** joue **U(up)**, la meilleure réponse de **B** est **L(left)**.
    - Si **B** joue **L (left)**, la meilleure réponse de **A** est **U(up)** → Donc (**U,L**) **est possible**.

2 Forme Normale : Définition Formelle et Notations

2.1 Définition

Un jeu en **forme normale** est décrit comme suit :

1. Un ensemble de N joueurs, J ≡ {1,2,…,N}
2. Chaque joueur i, i J a un ensemble d'actions (Strategies) A^i qui est l'ensemble de toutes les actions possibles pour i. Soit a^i A^i, une action (Strategie) particulière de A^i. On appelle a^i un résultat du jeu.
3. Chaque joueur a une fonction de payoff, Π^i qui assigne un nombre réel $\Pi^i(a)$, à chaque action du joueur i.

		Joueur B: L	Joueur B: R
Joueur A	U	(3,9)	(1,8)
	D	(0,0)	(2,1)

1) J= {Joueur A, Joueur B}

2) S_1 = $S_{Joueur\,A}$= {U, D}; S_2= $S_{Joueur\,B}$= {L,R}

3) Π^1 (U)= 3 ou 1 Π^1 (D)= 0 ou 2

Π^2 (L)= 9 ou 0 Π^2 (R)= 8 ou 1

5.2.5 Exercice 22 (Dilemme du prisonnier)

Enoncé

Deux suspects Ali et **Fahd** sont arrêtés par la police, mais la police manque de preuve pour les emprisonner. La police doit les faire avouer :

- Si les **deux avouent**, ils auront chacun **4 ans** de prison
- Si l'un **avoue** et l'autre **nie**,
    - Celui qui a **avoué** encourera **1 an de** prison
    - L'autre qui a **nié** encourera **10 ans** de prison
- Si les **deux nient**, ils auront chacun **2 ans** de prison

- **Questions :**

1) Donner l'ensemble de joueurs et l'ensemble de stratégies (actions)

2) Représenter le jeu sous forme normale

3) Jouer le jeu : Qu'il sera le résultat de ce jeu ? (Nombre d'années de prison) ? Peut-on l'améliorer ? Comment ?

- **Solution :**

1) L'ensemble de joueurs **J = {Ali, Fahd}**

L'ensemble de Stratégies (Actions) A**1 = A2 = {avoue, nie}**

2) La forme normale de ce jeu est :

		Fahd	
		avoue	nie
Ali	avoue	(4;4)	(1;10)
	nie	(10;1)	(2;2)

3) Le résultat de jeu est **(avoue, avoue)** avec un gain de (**4,4), oui** on peut l'améliorer si les deux prisonniers **coopèrent**

3 Information parfaite/imparfaite

On parle de jeu à information parfaite dans le cas de jeu sous forme extensive, où chaque joueur a une connaissance parfaite de toute l'histoire du jeu. : On dit alors qu'un jeu est à information complète si chaque joueur connaît lors de la prise de décision :

- Ses possibilités d'action
- Les possibilités d'action des autres joueurs
- Les gains résultants de ces actions

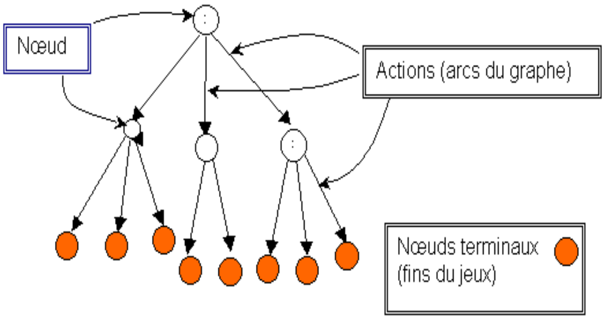
Un jeu à information incomplète est aussi à information imparfaite

- Les échecs sont à information complète et parfaite
- Le jeu de Poker : est à information incomplète et imparfaite

4 Forme extensive de jeu

La forme extensive : est un arbre (graphe connexe sans cycle) représentant les déroulements possibles du jeu.:

- à chaque **Sommet** non terminal est associé **un joueur** : arrivé à ce point du jeu c'est à son tour de jouer.
- Chaque **arc** représente chacune **des actions** (coups autorisés par la règle) que ce joueur peut prendre à ce point du jeu.
- à chaque **sommet terminal** correspond un résultat du jeu donné par vecteur des paiements (liste des gains attribués à chaque joueur).



La forme extensive de jeu prend deux représentations selon la nature séquentielle ou simultanée de jeu.

a- Jeu séquentiel

Un jeu séquentiel est une forme de jeu dans la théorie des jeux où les joueurs prennent leurs décisions tour à tour, en ayant connaissance des actions précédentes des autres joueurs. Cela implique une dimension temporelle importante, car chaque joueur choisit sa stratégie en fonction des actions précédentes et anticipe les réponses des autres, ce qui peut conduire à des résultats différents par rapport à un jeu simultané

b- Jeu simultané

Un jeu simultané est une forme de jeu dans la théorie des jeux où les joueurs prennent leurs décisions en **même temps**, sans connaître les choix des autres joueurs au moment de prendre leur décision. Chaque joueur choisit sa stratégie indépendamment des actions des autres, ce qui peut conduire à des situations d'équilibre où chaque joueur cherche à maximiser son gain compte tenu des actions potentielles des autres joueurs.

4.1) Exemple

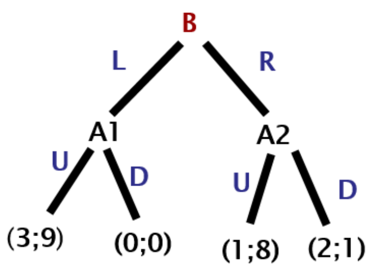
Considérons le jeu sous sa forme normale suivante :

		Bihi	
		L	R
Ali	U	(3,9)	(1,8)
	D	(0,0)	(2,1)

La représentation de ce jeu sous **forme extensive** peut prendre deux formes, selon la nature de jeu, est ce que le jeu est **séquentiel** ou **simultané**.

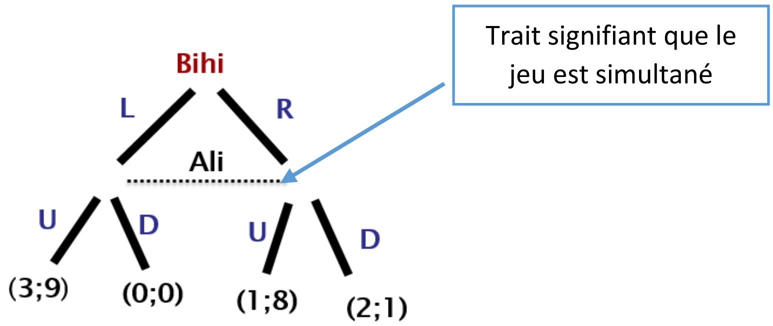
a- Le cas de jeu séquentiel (**Ali joue après Bihi)**

Dans ce cas, le jeu est **séquentiel** et **Ali** connait l'action de **Bihi**. La forme **extensive** du jeu sera comme suit :



d) Le cas de jeu **Simultané** (**Ali et Bihi jouent en meme temps)**

Dans ce cas ni **Ali** ni **Bihi** ne connait l'un l'action de l'autre et la **représentation extensive** devient comme suit :



Le trait « **……….** » Signifie que le jeu est **simultané**

5.2.6 Exercice 23 (papier, ciseau et caillou)

- **Enoncé** :

Il s'agit d'un jeu entre deux enfants **Ali** et **leila**, les deux choisissent un objet parmi les 3 suivant : **papier**, **ciseau** et **caillou**. Selon ces choix, soit l'enfant gagne le jeu soit il n y a pas de gagnant (s'ils choisissent le même objet).

Papier gagne contre caillou, **ciseau** gagne contre **papier** et **caillou** gagne contre **ciseau**. Soit **2 le gain** de l'enfant qui gagne, **0 le gain** de celui qui **perd et 1 le gain en cas d'égalité**.

Questions :

1) Donner l'ensemble de joueurs et l'ensemble de stratégies (actions) de chaque joueur.

2) Représenter le jeu simultané en forme normale

3) Représenter le jeu simultané en forme extensive

4) Même question que (3) si **Ali** triche et observe le choix de Leila avant de jouer

5) Même question que (3) si **Ali** n'observe le choix de Leila que s'elle choisit caillou

Solution : au cours de la séance de TD

Chapitre 7 : équilibre du jeu

Le chapitre sur l'équilibre des jeux est une plongée profonde dans les concepts fondamentaux de la théorie des jeux, où nous allons explorer la manière dont les joueurs parviennent à des décisions rationnelles dans des situations d'interaction stratégique. Au cœur de ce chapitre se trouve l'étude de la résolution d'un jeu et des équilibres qui en résultent.

Nous commencerons par examiner ce qui se passe dans un jeu en l'absence d'un équilibre, une situation complexe où les joueurs ne parviennent pas à des choix optimaux, conduisant à des résultats sous-optimaux. Cela nous conduira à explorer la notion de stratégie dominante et stratégie dominée, où nous analyserons comment les joueurs peuvent éliminer des options de leur ensemble de stratégies pour arriver à des décisions plus rationnelles.

Ensuite, nous aborderons la notion d'équilibre en stratégies dominantes, par Elimination Itérative des Stratégie dominée (EISD), qui représente une situation dans laquelle chaque joueur choisit sa stratégie dominante, ce qui constitue un équilibre. Nous examinerons également l'équilibre de Nash, une notion centrale de la théorie des jeux, où chaque joueur choisit sa stratégie en anticipant les actions des autres. Cet équilibre de Nash peut différer de l'EISD et peut donner lieu à des solutions plus riches et nuancées dans les jeux.

Ce chapitre fournira une base solide pour comprendre comment les joueurs parviennent à des décisions rationnelles et équilibrées dans des jeux complexes, en explorant ces notions essentielles de la théorie des jeux.

1 Résolution d’un jeu et Equilibre.

1.1 Résolution d’un jeu

La résolution d’un jeu est une méthode qu’on peut suivre pour vérifier est ce un jeu a un équilibre ou pas. Considérons deux joueurs A et B ayant respectivement des décisions stratégiques Ai et Bi i∊ {1,2…n} avec un gain pour chaque action. La résolution de ce jeu suit Les étapes de résolutions ci-dessous :

1. Identifier les décisions de A
   - ✓ Meilleure décision de A, comptenu la décision B1 de joueur B
   - ✓ Meilleure décision de A, compte tenu la décision B2 de joueur B,
   - ✓ ….
   - ✓ Meilleure décision de A, compte tenu la décision Bn de joueur B,
2. Identifier les décisions de B
   - ✓ Meilleure décision de B, compte tenu la décision A1 de joueur A
   - ✓ Meilleure décision de B, compte tenu la décision A2 de joueur A,
   - ✓ …..
   - ✓ Meilleure décision de B, compte tenu la décision An de joueur A,
3. On caractérise la solution du jeu, si elle existe

1.2 Exemple de la Résolution d'un jeu

Considérons le jeu dont la forme normale es ci-dessous

Matrice des gains		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	(2, 2)	(15,0)
	Q_e	(0,15)	(10,10)

La résolution de ce jeu selon les étapes citées dans la section précédente sera abordée comme suit :

1. Seules les décisions de A sont prises en compte

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	(2,)	(15,)
	Q_e	(0,)	(10,)

2. Seules les décisions de A sont retenues si B choisit **Qd**

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	(2,)	(,)
	Q_e	(0,)	(,)

3. On retient pour A la décision qui génère le plus gros gain c'est **Qd** marqué en * qui genre 2 (car **Qe** génère 0)

Qd>Qe

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	*(2, 2)	(15,0)
	Q_e	(0,15)	(10,10)

4. Seules les décisions de A sont prises en compte (on revient à l'étape 1 avec **Qe** pour **B** voir l'etape suivante)

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	*(2,)	(15,)
	Q_e	(0,)	(10,)

5. Seules les décisions de A sont retenues si B choisit **Qe**

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	*(2,)	(15,)
	Q_e	(0,)	(10,)

6. On retient pour **A** la décision qui génère le plus **gros gain (*) qui est Qd (car 15>10)**

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	*(2, 2)	*(15,0)
	Q_e	(0,15)	(10,10)

Maintenant qu'on a identifié les gains de A selon les deux actions de B on passe pour identifier les gains de B selon les action de A

7. Seules les décisions de B sont prises en compte

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	*(, 2)	*(,0)
	Q_e	(,15)	(,10)

8. Seules les décisions de B sont retenues si A choisit **Qd**

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	*(, 2)	*(,0)
	Q_e	(,)	(,)

9. On retient pour **B** la décision qui génère le plus gros gain (*) c'est **Qd** marqué en * qui genre 2 (car **Qe** génère 0)

 Qd>Qe (pour B si A choisi Qd)

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	*(2, 2)*	*(15,0)
	Q_e	(0,15)	(10,10)

10. Seules les décisions de B sont prises en compte (on revient à l'étape 7 avec **Qe** pour **A** voir l'étape suivante)

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	*(, 2)*	*(,0)
	Q_e	(,15)	(,10)

11. Seules les décisions de B sont retenues si A choisi **Qe**

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	*(,)*	*(,)
	Q_e	(,15)	(,10)

12. On retient pour **B** la décision qui génère le plus **gros gain (*) qui est Qe (car 15>10)**

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	*(,)*	*(,)
	Q_e	(,15)*	(,10)

Résultat :

- Un jeu a un équilibre quand il génère une **convergence** des décisions stratégiques

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	*(2, 2)*	*(15,0)
	Q_e	(0,15)*	(10,10)

Convergence
des décisions

- Le couple de stratégies **(Q_d;Q_d)** est la solution du jeu

1.3 Jeu en équilibre

Un jeu en équilibre se réfère à une situation dans la théorie des jeux où chaque joueur a choisi une stratégie de telle manière que, compte tenu des choix des autres joueurs, il n'a pas d'incitation à changer sa propre stratégie pour obtenir un meilleur résultat. En d'autres termes, un équilibre dans un jeu est un état où chaque joueur maximise son propre bénéfice, sachant que les autres joueurs font de même. Il existe plusieurs types d'équilibres,

Exemple : le jeu étudié dans l'exemple précédent est un jeu en équilibre, car les deux joueurs **convergent vers** la décisions stratégiques **(Qd,Qd)**

		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	*(2, 2)*	(15,0)
	Q_e	(0,15)	(10,10)

1.4 Jeu sans équilibre

Un jeu sans équilibre se réfère à une situation dans la théorie des jeux où, malgré une analyse approfondie des stratégies des joueurs, aucune combinaison de choix stratégiques ne conduit à un état stable ou équilibré. En d'autres termes, il n'existe aucune solution où chaque joueur est satisfait de sa stratégie, et aucune incitation à modifier ses choix pour obtenir un meilleur résultat. Les jeux sans équilibre peuvent être complexes et souvent imprévisibles, car les joueurs peuvent être pris dans des dilemmes ou des contradictions stratégiques, conduisant à des résultats sous-optimaux.

Exemple d'un jeu sans équilibre

Considérons un jeu sous la forme normale suivante :

		Joueur 2	
		S_1	S_2
Joueur 1	S_1	(0,10)	(10,0)
	S_2	(10,0)	(0,10)

En appliquant les étapes de la résolution de jeu on arrive à la situation suivante :

		Joueur 2	
		S_1	S_2
Joueur 1	S_1	(0,10)*	*(10,0)
	S_2	*(10,0)	(0,10)*

Il n'y a pas de **convergence de stratégie entre les deux joueurs, donc** ce jeu est un jeu est **sans équilibre**

1.4.1 Exercice 24 :

Faire la résolution des deux jeux ci-dessous en appliquant les étapes vues ci-dessus, et déterminer la nature de chacun selon jeu en équilibre ou jeu sans équilibre. Déterminer le couple de stratégies ou converent les deux joueurs si le jeu est en équilibre

Jeu 1
Dilemme de prisonniers

		Fahd	
		avoue	nie
Ali	avoue	(-4;-4)	(-1;-10)
	nie	(-10;-1)	(-2;-2)

Jeu 2

		Joueur B		
		X	Y	Z
Joueur A	X	3 , 6	5,7	4,5
	Y	5 ,1	6,2	6,1
	Z	6 , 0	8,9	3,6

2 Stratégie Dominante / Dominée

Dans le cadre de la théorie des jeux, la notion de stratégie dominante et dominée est cruciale pour comprendre comment les joueurs prennent des décisions dans des situations interactives. En explorant cette section, vous allez découvrir comment les acteurs rationalisent leurs choix stratégiques en évaluant les conséquences de leurs actions, tout en tenant compte des actions de leurs adversaires. Comprendre la dynamique des stratégies dominantes et dominées est essentiel pour anticiper le comportement des joueurs et optimiser ses propres décisions, que ce soit dans un jeu de société, dans le monde des affaires ou dans des scénarios plus complexes, tels que la négociation et la politique.

2.1 Notations

Un jeu en forme normale est décrit comme suit :

- ✓ Un ensemble de N joueurs, $J \equiv \{1,2,\ldots,N\}$
- ✓ Chaque joueur J a un ensemble de **m** Stratégies $S_J = \{s_{J1}, s_{J2}, ..s_{Jm}\}$
- ✓ Chaque joueur J a une fonction de **payoff(gain), Π^J** qui assigne un nombre réel $\Pi^J(S_J)$, à chaque **action** du joueur J.

Soient :

- s_i un profil de stratégies $\{s_{1J},\ldots..s_{mJ}\}$ / qlq soit i, $s_i \in$à S_J

- s_{-i} *(s indice –i)* le profil ***de*** stratégies autres que celles du joueur ***J*** :

  $s_{-i} = \{s_1, \ldots\ldots S_{i-1}, S_{i+1}, \ldots\ldots s_N\}$ (N est le nombre de joueur)

- On note S l'espace des stratégies ie : $S = S_1 x S_2 x \ldots\ldots x S_N$

Principe

Une stratégie **n'est jamais jouée** si une autre assure un gain meilleur dans tous les cas.

2.2 Définition :

- Une stratégie ***sd*** est (**strictement**) **dominée** :

pour le joueur **J**, s il existe une stratégie s_i telle que pour tous les gains générés par les stratégies autre que **sd** (qu on note par s_{-i}) on a :

$$\Pi^J(s_i, s_{-i}) > \Pi^J(sd, s_{-i})$$

- Une stratégie **sf** est **faiblement dominée**

Pour le joueur **i** s il existe une stratégie ***s'i*** telle que pour tous les gains correspondant à ***s-i on a***

$$\Pi^J(s_i, s_{-i}) \geq \Pi^J(sd, s_{-i})$$ (**sup. ou égal**)

a- Exemple

Soit le jeu sous forme normale ci-dessous

		J2	
		X	Y
J1	U	(5 ,3)	(4,2)
	V	(3,6)	(7, 1)

Un jeu en forme normale est décrit comme suit :

- ✓ Un ensemble de 2 joueurs, $J \equiv \{1,2\}$
- ✓ Chaque joueur ***J*** a un ensemble de **m** Stratégies $S_J = \{s_{J1}, s_{J2}, ..s_{Jm}\}$
  - $S_1 = \{U,V\}$
  - $S_2 = \{X,Y\}$
- ✓ Chaque joueur ***J*** a une fonction de **payoff(gain),** Π^J qui assigne un nombre réel $\Pi^J(S_J)$, à chaque **action** du joueur ***J***.
  - $\Pi^1(U, X) = 5$
  - $\Pi^2(U, X) = 3$
  - $\Pi^1(V, X) = 3$
  - $\Pi^2(V, X) = 6$
  - $\Pi^1(U, Y) = 4$
  - $\Pi^2(U, Y) = 2$
  - $\Pi^1(V, Y) = 7$
  - $\Pi^2(V, Y) = 2$
- ✓ On note S l'espace des stratégies : $S = S_1 x S_2$
- Pour le joueur **J2 on** a :

$\Pi^2(V, X) > \Pi^2(V,Y)$ et

$\Pi^2(U, X) > \Pi^2(U, Y)$

Donc

Y est une Stratégie dominée de J2

X est une Stratégie dominante de J2

- Pour le joueur **J1** on a :

$\Pi^1(U, X) > \Pi^1(V, X)$

$\Pi^1(U, Y) < \Pi^1(V, Y)$

On déduit que parfois > et autre < donc les stratégies de J_1 sont Incomparables : on dit qu'il n'y a pas de stratégie dominante ni Dominée pour J_1

2.3 Equilibre en Stratégie dominante

Une stratégie dominante est une stratégie qu'un joueur préfère, indépendamment des choix des autres joueurs. En d'autres termes, si un joueur a une stratégie dominante, il la choisira, peu importe ce que font les autres joueurs, car elle lui garantit un résultat optimal compte tenu de ses préférences.

Lorsqu'un jeu atteint l'équilibre en stratégie dominante, chaque joueur choisit sa stratégie dominante, et aucun joueur n'a d'incitation à changer de stratégie, car cela ne lui apporterait pas un meilleur résultat. Pour illustrer ce concept on propose l'exemple suivant :

Player 2

Player 1

	u	v	s
x	4,2	3,6	3,1
y	2,5	1,8	3,0
z	8,1	4,2	6,0

Analysons ce jeu :

Le signe % = par rapport à

Player 1

% U de Player 2 $\Pi^1(z, u) > \Pi^1(y, u)$ $\Pi^1(z, u) > \Pi^1(x, u)$

% V de Player 2 $\Pi^1(z, v) > \Pi^1(y, v)$ $\Pi^1(z, v) > \Pi^1(x, v)$

% S de Player 2 $\Pi^1(z, s) > \Pi^1(y, s)$ $\Pi^1(z, s) > \Pi^1(x, s)$

➔ La stratégie Z domine pour Player 1

Player 2

Player 1

	u	v	s
x	4,2	3,6	3,1
y	2,5	1,8	3,0
z	8,1	4,2	6,0

Player 2

% X de Player 1 $\Pi^2(x,v) > \Pi^2(x,u)$ $\Pi^2(x,v) > \Pi^2(x,s)$

% Y de Player 1 $\Pi^2(y,v) > \Pi^2(y,u)$ $\Pi^2(y,v) > \Pi^2(y,s)$

% Z de Player 1 $\Pi^2(z,v) > \Pi^2(z,u)$ $\Pi^2(z,v) > \Pi^2(z,s)$

➔ La stratégie V domine pour Player 2

Player 2

Player 1

	u	v	s
x	4,2	3,6	3,1
y	2,5	1,8	3,0
z	8,1	4,2	6,0

- **Résultat de l'analyse**

L'équilibre en Stratégie Dominante existe et vaut (z,v) avec une utilité de ➔ (4,2)

Player 2

Player 1

	u	v	s
x	4,2	3,6	3,1
y	2,5	1,8	3,0
z	8,1	4,2	6,0

En d'autre terme, l'équilibre en stratégies dominées est l'intersection des deux décessions stratégiques dominantes pour chaque joueur **Player 1** et **Player 2** qui est dans ce cas **(Z,V)** qui donne un gain de **4** pour Player1 et un gain de 2 pour **Player 2** . $\Pi^{player1}(Z, V)=4$ et $\Pi^{player2}(Z, V)=2$

3 Equilibre en Elimination Itérative de Stratégies Dominées (EISD)

L'équilibre en Élimination Itérative de Stratégies Dominées (EISD) offre un outil puissant pour simplifier l'analyse des jeux et identifier des solutions équilibrées. L'EISD repose sur le principe que, dans certaines situations de jeu, les joueurs peuvent éliminer de manière logique et rationnelle certaines stratégies de leur ensemble de choix.

Imaginez un jeu où chaque joueur doit choisir parmi plusieurs stratégies. L'EISD consiste à éliminer progressivement les stratégies qui ne sont pas optimales, c'est-à-dire celles qui ne sont pas les meilleures réponses à ce que font les autres joueurs. Ce processus itératif de réduction conduit finalement à un ensemble de stratégies restantes, où chaque joueur a une stratégie dominante.

Au cours de cette section, nous examinerons de près le concept d'EISD, comment il est appliqué pour simplifier l'analyse des jeux, et comment il peut être utilisé pour identifier des équilibres potentiels dans des jeux complexes.

3.1 L’algorithme EISD

L'algorithme de l'Élimination Itérative de Stratégies Dominées (EISD) est un processus itératif qui permet de simplifier un jeu en éliminant les stratégies dominées jusqu'à ce qu'un équilibre émerge. Voici comment fonctionne cet algorithme :

Étape 1 - Définition du jeu initial : Tout d'abord, le jeu est défini, y compris les joueurs, leurs ensembles de stratégies, et les gains associés à chaque combinaison de stratégies.

Étape 2 - Identification des stratégies dominées : Pour chaque joueur, l'algorithme commence par examiner si une stratégie donnée est dominée par une autre stratégie. Une stratégie est dite dominée si elle ne permet pas au joueur d'obtenir un meilleur résultat que celui obtenu en choisissant une autre stratégie, quelle que soit la stratégie choisie par les autres joueurs.

Étape 3 - Élimination des stratégies dominées : Une fois qu'une stratégie dominée est identifiée, elle est éliminée de l'ensemble de stratégies du joueur. Cela réduit le nombre d'options dont dispose le joueur pour prendre des décisions.

Étape 4 - Répétition de l'analyse : Les étapes 2 et 3 sont répétées jusqu'à ce qu'aucune stratégie dominée ne puisse être identifiée pour aucun joueur. À ce stade, le jeu est simplifié autant que possible en termes de stratégies dominées.

Étape 5 - Analyse des équilibres restants : Une fois que toutes les stratégies dominées ont été éliminées, on examine les stratégies restantes pour chaque joueur. Ces stratégies, souvent appelées stratégies non dominées ou stratégies candidates, représentent les choix potentiels des joueurs à l'équilibre.

Étape 6 - Identification de l'équilibre : Si un ensemble de stratégies est identifié où chaque joueur a une stratégie dominante, alors cet ensemble représente un équilibre en stratégies dominantes (EISD). Cela signifie que chaque joueur a rationalisé ses choix de manière à maximiser ses gains compte tenu des stratégies des autres joueurs, et aucun joueur n'a d'incitation à changer de stratégie.

Il est important de noter que l'EISD n'est qu'une approche parmi plusieurs pour analyser les équilibres dans les jeux. Dans certains cas, il peut ne pas exister d'EISD, et d'autres formes d'équilibre, comme l'équilibre de Nash, peuvent être plus appropriées pour décrire le comportement des joueurs. L'EISD est particulièrement utile pour simplifier des jeux complexes en éliminant des stratégies non optimales, mais il ne garantit pas toujours la présence d'un équilibre dans tous les jeux.

3.2 Exemple Haut du formulaire

Considérons le jeu ci-dessous

		Joueur 2	
		u	v
Joueur 1	x	4,2	3,1
	y	2,5	9,0

Application de l'algorithme

Étape 1 - Définition du jeu initial

Joueur 1

$\Pi^1(X,U)=4$, $\Pi^1(X,V)=3$

$\Pi^1(Y,U)=2$, $\Pi^1(Y,V)=9$

$\Pi^2(X,U)=2$, $\Pi^2(X,V)=1$

Joueur 2

$\Pi^2(X,U)=5$, $\Pi^2(X,V)=0$

		Joueur 2	
		u	v
Joueur 1	x	4,2	3,1
	y	2,5	9,0

Étape 2 -- Identification des stratégies dominées

Joueur 1

$\Pi^1(X,U)=4 > \Pi^1(Y,U)=2$

$\Pi^1(X,V)=3 < \Pi^1(Y,V)=9$

Pas de stratégie dominée pour j1

Joueur 2

$\Pi^2(X,U)=2 > \Pi^2(X,V)=1$

$\Pi^2(X,U)=5 > \Pi^2(X,V)=0$

V est une stratégie dominée pour j2

Étape 3 - Élimination des stratégies dominées

Joueur 1

PAS DE STRATEGIE DOMINEE POUR J1

Rien n'est a éliminer

Joueur 2

V EST UNE STRATEGIE DOMINEE POUR J2

La stratégie V est à ELIMINER pour J2

		Joueur 2	
		u	~~v~~
Joueur 1	x	4,2	~~3,1~~
	y	2,5	~~9,0~~

Étape 4 - Répétition de l'analyse : Les étapes 2 et 3 sont répétées

Joueur 1

V ne sera jamais jouée par J2 $\Pi^2(X,U)=2 > \Pi^2(Y,U)=1$

La stratégie Y est à ELIMINER pour J1

		Joueur 2	
		u	~~v~~
Joueur 1	x	4,2	~~3,1~~
	y	~~2,5~~	~~9,0~~

	Joueur 2	**N' a qu'une seul stratégie qui reste à savoir U**	

Joueur 2

Joueur 1	u	~~v~~
x	4,2	~~3,1~~
y	~~2,5~~	~~9,0~~

Étape 5 - Analyse des équilibres restants :	**Joueur 1 et 2**	toutes les stratégies dominées ont été éliminées,	les choix potentiels des joueurs à l'équilibre est le candidat **(X,U)**
Étape 6 - Identification de l'équilibre	**Joueur 1 et 2**	L'ensemble de stratégies est identifié où chaque joueur a une stratégie dominante,	**Résultat de jeu est ; (X,U)** représente un équilibre en stratégies dominantes (EISD).

3.3 Exemple 2 :

Soit le jeu suivant :

Joueur 2

Joueur 1	u	v	w
x	3,6	7,1	4,8
y	5,1	8,2	6,1
z	6,0	6,2	3,2

1er Itération : (Joueur 1) : On voit bien que X est une stratégie dominée par Y pour le Joueur 1 quel que soit la décision de Joueur 2 → on élimine donc la stratégie X pour le Joueur 1 de ce jeux, il en résulte donc ça :

Joueur 2

Joueur 1	u	v	w
~~x~~	~~3,6~~	~~7,1~~	~~4,8~~
y	5,1	8,2	6,1
z	6,0	6,2	3,2

2eme Itération (Joueur 2); Maintenait on constat que U est une stratégie dominée par V (encore faiblement dominée par **W)** pour le **Joueur 2** quel que soit les décisions restantes de **Joueur 1**→ on élimine donc la stratégie U pour le Joueur 2 de la représentation précédente, il en résulte donc ça :

Joueur 2

Joueur 1

	u	v	w
x	~~3.6~~	~~7,1~~	~~4.8~~
y	~~5,1~~	8,2	6,1
z	~~6,0~~	6,2	3,2

3eme Itération (Joueur 1); on constat que Z est une stratégie dominée par Y pour le **Joueur 1** quel que soit les décisions restantes de **Joueur 2**→ on élimine donc la stratégie Z pour le Joueur 1 de la représentation précédente, il en résulte donc ça :

Joueur 2

Joueur 1

	u	v	w
x	~~3.6~~	~~7,1~~	~~4.8~~
y	~~5,1~~	8,2	6,1
z	~~6,0~~	~~6,2~~	~~3,2~~

4eme Itération (Joueur 2); on constat que V est une stratégie dominée par W pour le **Joueur2** quel que soit les décisions restantes de **Joueur 1**→ on élimine donc la stratégie V pour le Joueur2 de la représentation précédente, il en résulte donc ça :

Joueur 2

Joueur 1

	u	v	w
x	~~3.6~~	~~7,1~~	~~4.8~~
y	~~5,1~~	8,2	~~6,1~~
z	~~6,0~~	~~6,2~~	~~3,2~~

- Toutes les stratégies dominées ont été éliminées,

Résultat de jeu est ;

(Y, V) représente un équilibre en stratégies dominantes (EISD).

		Joueur 2		
		u	v	w
Joueur 1	x	~~3.6~~	~~7,1~~	~~4.8~~
	y	~~5,1~~	8,2	~~6,1~~
	z	~~6,0~~	~~6,2~~	~~3,2~~

3.1) Remarques

- ❖ Un jeu est dit résolvable par élimination itérative des stratégies dominées, si on obtient un unique profil en éliminant successivement des stratégies (strictement) dominées.
- ❖ Les profils obtenus aprés élimination itérative des stratégies (*strictement*) dominées (EISSD) ne dépendent pas de l'ordre choisi pour l'élimination des stratégies.
- ❖ Par contre, on peut obtenir des profils différents lorsque l'on choisit des ordres différents pour l'élimination itérative de stratégies *faiblement* dominées (EISFD).
- ❖ Les résultats obtenus par EISSD sont donc plus robustes que ceux obtenus par EISFD.
- ❖ Problème majeur de cette méthode: tous les jeux ne sont pas résolvable par EISD !

3.3.1 Exercice 25

a- Enoncé

Pour les 3 jeux sous forme normale ci-dessous

1) Y a-t-il un équilibre en stratégie dominante
2) Existe-t-il un équilibre en EISD (Elimination Itérative de Stratégie Dominées)

Jeu 1
Dilemme de prisonniers

		Fahd	
		avoue	nie
Ali	avoue	(-4;-4)	(-1;-10)
	nie	(-10;-1)	(-2;-2)

Jeu 2
2 sociétés

		St Y	
		bas	Haut
St X	bas	40,30	45,10
	Haut	20,60	50,50

Jeu 3

		Joueur B		
		X	Y	Z
Joueur A	X	3 , 6	5,7	4,5
	Y	5 ,1	6,2	6,1
	Z	6 , 0	8,9	3,6

b- Correction :

- **Jeu1 : Dilemme de prisonniers**

1) Vérifions qu'il y a un équilibre en **stratégie dominante**

Pour Ali ➔ $(\Pi^{ali}(\boldsymbol{avoue}, \boldsymbol{avoue}) > \Pi^{ali}(\boldsymbol{nie}, \boldsymbol{avoue})$

Et $(\Pi^{ali}(\boldsymbol{avoue}, \boldsymbol{nie}) > \Pi^{ali}(\boldsymbol{nie}, \boldsymbol{nie})$

		Fahd	
		avoue	nie
Ali	avoue	(–4;–4)	(–1;–10)
	nie	(–10;–1)	(–2;–2)

Donc **avoue** est une **stratégie dominante** pour **Ali**

Pour Fahd ➔ $(\Pi^{fahd}(\textit{avoue}, \textit{nie})) > \Pi^{fahd}(\textit{nie}, \textit{nie})$

Et $(\Pi^{fahd}(\textit{avoue}, \textit{avoue}) > \Pi^{fahd}(\textit{nie}, \textit{avoue})$

		Fahd	
		avoue	nie
Ali	avoue	(–4;–4)	(–1;–10)
	nie	(–10;–1)	(–2;–2)

Donc **avoue** est une **stratégie dominante** pour **Fahd**

On réalise l'intersection des deux représentations :

		Fahd	
		avoue	nie
Ali	avoue	(–4;–4)	(–1;–10)
	nie	(–10;–1)	(–2;–2)

Résultat, il **y a un équilibre en** stratégie dominante ➔ **(avoue,avoue)**

2) Vérifions qu'il y a un équilibre par EISSD

		Fahd	
		avoue	nie
Ali	avoue	(–4;–4)	(–1;–10)
	nie	(–10;–1)	(–2;–2)

Pour Ali ➔ $(\Pi^{ali}(\textit{nie}, \textit{nie}) < \Pi^{ali}(\textit{avoue}, \textit{nie})$

Et $\Pi^{ali}(\textit{nie}, \textit{avoue}) < (\Pi^{ali}(\textit{avoue}, \textit{avoue})$

Donc **nie** est une **stratégie dominée** pour **Ali** → **on va l'éliminer**, il en résulte la représentation suivante :

		Fahd	
		avoue	nie
Ali	avoue	(–4;–4)	(–1;–10)
	~~nie~~	~~(–10;–1)~~	~~(–2;–2)~~

Pour Fahd ➔ $(\Pi^{fahd}(\textit{nie}, \textit{nie}) < \Pi^{fahd}(\textit{avoue}, \textit{nie})$

Et $(\Pi^{fahd}(\textit{nie}, \textit{avoue}) < \Pi^{fahd}(\textit{avoue}, \textit{avoue})$

Donc **nie** est une **stratégie dominée pour Fahd** → **on va l'éliminer**, il en résulte la représentation suivante :

		Fahd	
		avoue	~~nie~~
Ali	avoue	(−4;−4)	~~(−1;−10)~~
	~~nie~~	~~(−10;−1)~~	~~(−2;−2)~~

Résultat , il y **a un équilibre** en **EISD** ➔ (**avoue,avoue**)

Remarque : un équilibre en Stratégie **Dominante est** un Equilibre en EISSD, mais **l'inverse** n'est **pas vrai**

- **Jeu 2 : 2 sociétés**

		St Y	
		bas	Haut
St X	bas	40,30	45,10
	Haut	20,60	50,50

1) Vérifions qu'il y a un équilibre en **stratégie dominante**

Pour St X ➔ $(\Pi^{stx}(\textit{bas}, \textit{bas}) = 40) > (\Pi^{stx}(\textit{haut}, \textit{bas}) = 20)$

et $(\Pi^{stx}(\textit{bas}, \textit{haut}) = \mathbf{45}) < (\Pi^{stx}(\textit{haut}, \textit{haut}) = \mathbf{50})$

Alors : Les deux stratégies **sont incomparables**

Il n'y a pas une **stratégie dominante** pour **St X**. Donc on s'arrête là et on annonce qu'il n'y a pas **d'équilibre en stratégie dominante** pour ce jeu,

Mais à titre indicatif on peut vérifier si **St Y** a une **stratégie dominée**

➔ $(\Pi^{sty}(\textit{bas}, \textit{bas}) = 30) > (\Pi^{sty}(\textit{haut}, \textit{bas}) = 10)$

et $(\Pi^{sty}(\textit{bas}, \textit{haut}) = 60) > \Pi^{sty}(\textit{haut}, \textit{bas}) = 50)$

Donc **bas** est une **stratégie dominante** pour **St Y**

Résultat, il n'y a pas **d'équilibre en stratégie dominante** pour ce jeu, car, il n'y a pas une **stratégie dominante** pour **St X**

2) Vérifions qu'il y a un équilibre en **EISSD.**

		St Y	
		bas	Haut
St X	bas	40,30	45,10
	Haut	20,60	50,50

Pour St Y ➔ $((\Pi^{sty}(\boldsymbol{haut}, \boldsymbol{bas})=10) < \Pi^{sty}(\boldsymbol{bas}, \boldsymbol{bas}) = 30)$

et $(\Pi^{sty}(\boldsymbol{haut}, \boldsymbol{bas}) = 50) < \Pi^{sty}(\boldsymbol{bas}, \boldsymbol{haut}) = 60)$

Donc **haut** est une **stratégie dominée** pour **St Y,** cette stratégie sera éliminée de l'ensemble des stratégies de St Y et le jeu devient :

		St Y	
		bas	~~Haut~~
St X	bas	40,30	~~45,10~~
	Haut	20,60	~~50,50~~

Pour St X ➔ $((\Pi^{stx}(\boldsymbol{haut}, \boldsymbol{bas}) = 20 < \Pi^{stx}(\boldsymbol{bas}, \boldsymbol{bas}) = 40))$

(Rationalité **St Y ne** jouera jamais Haut**)**

Donc **Haut** est **strictement dominé** par **bas** pour **St X,** la représentation de jeu devient

		St Y	
		bas	~~Haut~~
St X	bas	40,30	~~45,10~~
	~~Haut~~	~~20,60~~	~~50,50~~

Résultat : l'équilibre **par EISD pour** ce jeu est **(bas, bas)**

Jeu 3 :

1) Vérifions qu'il y a un équilibre en **stratégie dominante**

		Joueur B		
		X	Y	Z
Joueur A	X	3 , 6	5,7	4,5
	Y	5 ,1	6,2	6,1
	Z	6 , 0	8,9	3,6

Pour le joueur A ➔ $(\Pi^{A}(X, X) = 3) < (\Pi^{A}(Y, X) = 20) < \Pi^{A}(Z, X) = 6)$

et $(\Pi^{A}(X, Z) = 4) > (\Pi^{A}(Z, Z) = 3))$

Les trois stratégies sont Incomparables, pas la peine de vérifier pour le joueur B, on déduit donc qu'il n'y a pas d'équilibre en **stratégie dominante pour ce jeu**

2) Vérifions qu'il y a un équilibre pour ce jeu en procédant par EISD

Joueur A \ Joueur B		X	Y	Z
	X	3 , 6	5,7	4,5
	Y	5 ,1	6,2	6,1
	Z	6 , 0	8,9	3,6

- A partir de la représentation sous forme normale ci-dessous de **jeu 3** on a :
- **Iteration1) Pour le Joueur A:** ➔ $(\Pi^A(Y, X) = 5) > (\Pi^A(X, X) = 3)$

$(\Pi^A(Y, Y) = 6) > (\Pi^A(X, Y) = 5)$

$(\Pi^A(Y, Z) = 6) > (\Pi^A(X, Z) = 4)$

Donc la stratégie **X** est dominée par **Y** pour le joueur **A**, par conséquence la stratégie **X** sera éliminée de l'ensemble des stratégies de **joueur A**, la représentation de jeu devient

Joueur A \ Joueur B		X	Y	Z
	~~X~~	~~3 , 6~~	~~5,7~~	~~4,5~~
	Y	5 ,1	6,2	6,1
	Z	6 , 0	8,9	3,6

- **Iteration2) Pour le Joueur B:** ➔ $(\Pi^B(Y, Y) = 1) > (\Pi^B(X, Z) = 1)$

$(\Pi^B(Z, Y) = 9) > (\Pi^B(Z, Z) = 6)$

Donc la stratégie **Z** est dominée par **Y** pour le joueur **B**, par conséquence la stratégie **Z** sera éliminée de l'ensemble des stratégies de **joueur B**, la représentation de jeu devient

Joueur A \ Joueur B		X	Y	~~Z~~
	~~X~~	~~3 , 6~~	~~5,7~~	~~4,5~~
	Y	5 ,1	6,2	~~6,1~~
	Z	6 , 0	8,9	~~3,6~~

- **Itération 3) Pour le Joueur A:** ➔ $(\Pi^A(Z, X) = 6) > (\Pi^A(Y, X) = 5)$

$(\Pi^A(Z, Y) = 8) > (\Pi^A(Y, Y) = 6)$

Donc la stratégie **Y** est dominée par **Z** pour le joueur **A**, par conséquence la stratégie **Y** sera éliminée de l'ensemble des stratégies de **joueur A**, la représentation de jeu devient

Joueur B

Joueur A	X	Y	Z
X	3 , 6	5,7	4,5
Y	5,1	6,2	6,1
Z	6 , 0	8,9	3,6

Itération 4 : Pour le Joueur B: ➔ $(\Pi^{B}(Z\,,\,Y)=9)$ > $(\,\Pi^{A}(Z\,,X\,)=0\,)$

Donc la stratégie **X est** dominée par **Y** pour le joueur **B**, par conséquence la stratégie **X** sera éliminée de l'ensemble des stratégies de **joueur B**, la représentation de jeu devient

Joueur B

Joueur A	X	Y	Z
X	3 , 6	5,7	4,5
Y	5,1	6,2	6,1
Z	6 , 0	8,9	3,6

Résultat : Existence de l'Equilibre en EISD → (z,y)

Chapitre 8 : équilibre de NASH

L'équilibre de Nash est une notion clé qui explore comment les acteurs rationnels prennent des décisions dans un contexte où leurs choix dépendent des décisions des autres. Ce chapitre vous guidera à travers les principes de base de l'équilibre de Nash, vous montrant comment les stratégies des joueurs interagissent pour créer des situations stables, où aucun joueur n'a intérêt à dévier de sa stratégie actuelle. Vous découvrirez comment cet équilibre est omniprésent dans des situations du quotidien, du dilemme du prisonnier à la concurrence entre entreprises, en passant par les négociations internationales. En comprenant l'équilibre de Nash, vous acquerrez un nouvel éclairage sur la manière dont les individus et les organisations prennent des décisions stratégiques, contribuant ainsi à élargir vos horizons sur la science de la décision.

1 Définitions, Notations et Propriétés

1.1 Jeu en stratégies pures et Jeu en stratégies mixtes

Un jeu en **stratégies pures**, en termes de théorie des jeux, se réfère à une situation où chaque joueur choisit une stratégie bien définie et spécifique, sans aucune composante de hasard ou d'incertitude. Dans ce contexte, les stratégies pures sont des actions déterministes et non aléatoires que les joueurs prennent pour atteindre un objectif donné.

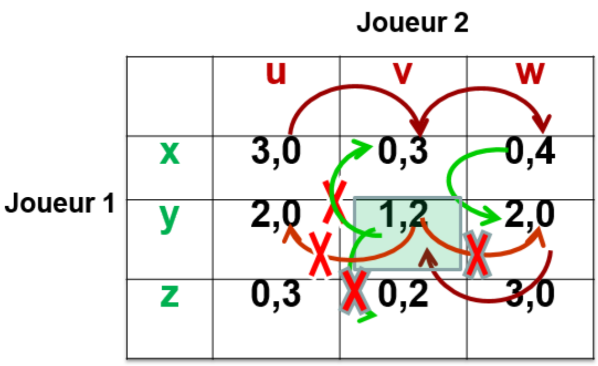
Un jeu en stratégies pures se distingue des **jeux en stratégies mixtes**, où les joueurs peuvent avoir recours à des stratégies probabilistes, introduisant ainsi une dimension aléatoire dans leurs décisions. Les jeux en stratégies pures sont plus simples à analyser, car les joueurs optent pour des actions claires et prévisibles, ce qui permet de déterminer plus facilement les équilibres de Nash, les solutions optimales, et les résultats du jeu.

1.2 L'équilibre de Nash

Une situation (**Profil**) du jeu où chaque stratégie est la meilleure réponse à l'autre est un **équilibre de Nash**. Autrement, **L'équilibre de Nash** est une situation où aucun joueur ne peut améliorer sa situation en changeant unilatéralement de stratégie, compte tenu des décisions des autres joueurs

a- Exemple

- Les flèches en vers sont relatives aux actions de **joueur1**
- Les flèches en rouge sont relatives aux actions de **joueur2**



Le profil **(X, U)** ne convient pas au joueur 2 car il peut améliorer son gain en **2** au lieu de **0** si il joue **V** au lieu de **U** ou **W pour** qu il passe à un gain de **3**, une fois on est dans le profil **(X,W)** le **joueur 1** à son tour peut améliorer son score de **0** à **2** s'il jeu y au lieu de x et ainsi de suite, dans ces situation on dit que le jeu n'est pas stable et les joueur peuvent toujours améliorer leur gain on changeant leur stratégie. Jusqu'à ce que **les deux joueurs choisissent le** profil **(Y, V)** avec un gain de **1** pour le **joueur 1** et **2** pour le **joueur2.**

Examinons ce profil (Y, V) :

- **Si le joueur1** décide de changer sa stratégie **de Y à Z ou à X** il va avoir **0, dans les deux cas, donc il n'a pas intérêt à le faire**
- **Si le joueur2** décide de changer sa stratégie **de V vers U ou W** il va avoir **aussi 0, dans les deux cas, donc il n'a pas intérêt à le faire**
- Une fois on est dans le profil **(Y, V)** aucun joueur n'a intérêt à changer sa stratégie,
- **Conclusion** : le profil **(Y, V)** est un **équilibre de Nash** pour ce jeu.

b- Formulation de l'équilibre de Nash

Un **éqilibre de Nash** est un profil de stratégies $\mathbf{s}^* = \{ \mathbf{s}^*_1, \ldots, \mathbf{s}^*_n \}$ tel que pour tout joueur $\boldsymbol{i}$, pour toute stratégie $\mathbf{s}^J \in \mathbf{s}_i$: $\Pi^i(s^*_i, s^*_{-i}) \geq \Pi^i(\mathbf{s}^J, s^*_{-i})$

1.3 Propriétés

✓ Un profil (unique) obtenu par élimination itérative de stratégies (**strictement**) dominées (**EISSD**) est un équilibre de Nash (et c'est le seul équilibre du jeu).

- ✓ Un jeu (en **stratégies pures**) peut **avoir plusieurs équilibres** de Nash, comme il peut aussi n'en **avoir aucun** !
- ✓ Deux équilibres de Nash s^* et s^{j*} sont *équivalents* s'ils donnent la même utilité à tous les joueurs, i.e. **Pour tout** $i \in N$ on a $\Pi^i(s^*) = \Pi^i(s^{j*})$

a- Exemple 1 : Jeu à 2 équilibres de Nash

Dans le jeu ci-dessous il y a **deux** équilibres de **Nash** à savoir :

(U, L) avec un payoff de (**3,9**) et (D, R) **avec** un payoff de **(2,1)**

Joueur B

Joueur A		L	R
	U	(3,9)	(1,8)
	D	(0,0)	(2,1)

(U,L) et **(D,R)** sont des équilibres de Nash pour ce jeu. Mais, lequel va apparaître ?

Nous remarquons que **(U,L)** est préféré à **(D,R)** par les deux joueurs. Pour autant est-ce que **(U,L)** va apparaître ?

Question : comment choisir un équilibre particulier lorsqu'il y en a **plusieurs ?**

Réponse : Lorsqu'un jeu contient deux équilibres de Nash, on parle d'une situation d'équilibre multiple. Dans de tels cas, les joueurs peuvent avoir plusieurs stratégies optimales, ce qui rend la prise de décision plus complexe.

Il existe généralement deux approches pour déterminer laquelle de ces stratégies équilibrées est la plus plausible :

1. **La stabilité** : Les joueurs peuvent opter pour l'équilibre de Nash qui semble le plus stable, c'est-à-dire celui où aucune des parties n'a une forte incitation à dévier de sa stratégie. Cette approche est basée sur l'idée que si un équilibre est plus stable, il est plus susceptible de se produire dans la pratique.
2. **La convention :** Les joueurs peuvent convenir à l'avance d'une stratégie particulière en cas de multiples équilibres de Nash. Cette approche peut découler de négociations, d'accords ou de règles établies à l'avance pour résoudre les situations d'équilibre multiple.

La décision de choisir l'un des équilibres de Nash dépend souvent du contexte du jeu, des informations dont disposent les joueurs, de leurs préférences, et de leur capacité à coordonner ou à communiquer. Il est également possible que les joueurs expérimentent différents équilibres dans des situations réelles pour voir lequel prévaut. Dans tous les cas, la théorie des jeux fournit un cadre pour analyser et comprendre ces situations d'équilibre multiple et leurs implications.

b- Exemple 2 : Equilibre de Nash avec des gains non Optimaux

Considérons le jeu de duopole ayant la forme normale suivante :

Matrice des gains		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	$\boxed{2,2}$	15,0
	Q_e	0,15	10,10

Considérons le profil **(Qd,Qd)** . Un joueur peut-il seul améliorer sa position ?

L'entreprise A ?   **NON**

L'entreprise B ?   **NON**

Matrice des gains		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	$\boxed{2,2}$	15,0
	Q_e	0,15	10,10

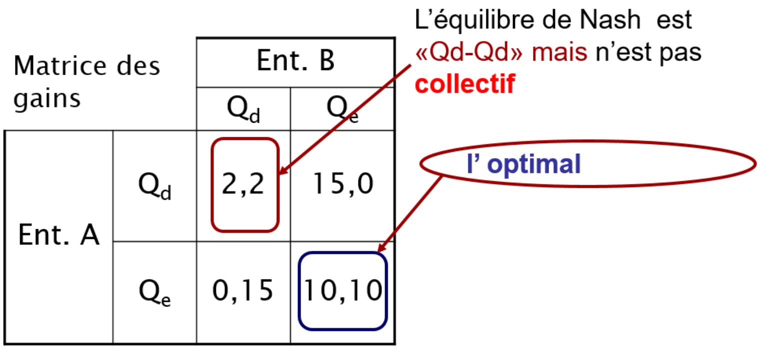
Puisque aucun joueur ne peut améliorer sa situation, comme le montre le dessin ci-dessous, aucune des deux entreprise **Ent A** et **EntB** n'a intérêt à changer sa stratégie alors **(Qd,Qd)** est un **équilibre de Nash.**

1.4 Equilibre de Nash non collectif et Jeu de coordination

Revenons à l'exemple précèdent

Matrice des gains		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	$\boxed{2,2}$	15,0
	Q_e	0,15	10,10

L'équilibre de Nash est **(Qd-Qd)** mais il réalise un gain **(2,2)** alors que le profil **(Qe-Qe)** réalise un gain **(10,10)** optimal, largement supérieur que celui de l'équilibre de Nash.



Dans ce cas **(Qd-Qd) est un équilibre de Nash** non collectif

Remarquons que puisque les gains en cas d'entente sont supérieurs aux gains sans entente, il s'agit **d'un jeu de *coordination***, En effet si les joueurs coordonnent entre eux ils **optimiseront leurs gains**

a- Equilibre de Nash non collectif

Un équilibre de Nash non collectif est une situation dans laquelle les joueurs d'un jeu, bien que chacun suive une stratégie qui est optimale pour lui-même, n'aboutissent pas nécessairement à une solution optimale pour l'ensemble du groupe ou de la société. Cela signifie que chaque joueur, en fonction de ses propres intérêts, a choisi la meilleure option, mais l'ensemble de ces choix ne conduit pas à la meilleure solution globale.

Ce type d'équilibre de Nash met en évidence un conflit potentiel entre les intérêts individuels et l'intérêt collectif. Il est souvent associé à des situations de dilemme du prisonnier et d'autres problèmes similaires où la coopération entre les joueurs pourraient conduire à de meilleurs résultats pour tout le monde, mais où la nature égoïste des joueurs les pousse à des choix sous-optimaux.

b- Jeu de coordination

Un jeu de coordination est un type de jeu en théorie des jeux dans lequel les joueurs doivent choisir leurs actions de manière à ce que le résultat collectif soit optimal pour l'ensemble du groupe. Dans un jeu de coordination, les joueurs cherchent à aligner leurs choix sur ceux des autres pour maximiser leur gain ou leur bénéfice collectif, plutôt que de maximiser leurs gains individuels.

L'essence d'un jeu de coordination réside dans la nécessité de coopération et de communication entre les joueurs pour atteindre un équilibre. Typiquement, il existe plusieurs équilibres de Nash dans de tels jeux, mais l'important est que tous les joueurs parviennent à choisir une stratégie commune pour atteindre un résultat préférable.

Les jeux de coordination se retrouvent dans divers contextes de la vie réelle, comme la décision de choisir un réseau social particulier, de convenir d'un lieu de rendez-vous, ou de décider de normes industrielles, où la conformité aux standards en vigueur peut conduire à une meilleure efficacité et à des avantages mutuels. La compréhension des jeux de coordination est essentielle pour analyser les situations de coordination et de coopération dans lesquelles les individus doivent prendre des décisions interdépendantes pour atteindre des résultats optimaux.

Dans l'exemple précèdent si les deux sociétés coopèrent et communiquent entre elles elles optimiseront leur gains et le jeu soit (**Qe,Qe**) qui sera un **équilibre provisoire et non stable.**

Matrice des gains		Ent. B	
		Q_d	Q_e
Ent. A	Q_d	2,2	15,0
	Q_e	0,15	10,10

l' optimal

1.4.1 Exercice : Poule mouillée

Soit le jeu suivant connu sous le nom de « Poule mouillée »

		A	
		Coopère	Trahit
B	Coopère	(6,6)	(1,10)
	Trahit	(10,1)	(-20,-20)

Question :

Ce Jeu a t- il l'Equilibre **de Nash** ?

Réponse :

Oui : 2 Equilibres de Nash :

(Coopère, Trahir) avec un gain de **(10,1)** ou **(Trahir, Coopère)** avec un gain **(1,10)**

Dans **ces deux équilibres de Nash,** toujours un joueur **se trouve défavorisé** devant l'autre mais si on choisit tous les deux **(Coopère, Coopère)** après **discussion et coordination**, **même** si ce **n'est pas un équilibre de Nash** on peut optimiser nos gains collectifs.

3) Equilibre de Nash et Jeu séquentiel

Dans les deux exemples précédents les joueurs jouent simultanément, alors qu'Il existe des jeux où les joueurs jouent l'un après l'autre : **jeux séquentiels**.

- Le joueur qui joue en premier est le **leader**,
- Celui qui joue en deuxième est le **follower**.

Note :

- Parfois, un jeu a **plusieurs équilibres de Nash** et il est **difficile** de savoir lequel va sortir du jeu…
- En revanche, quand **un jeu est séquentiel**, il est **possible** de dire **quel équilibre de Nash va sortir du jeu**.

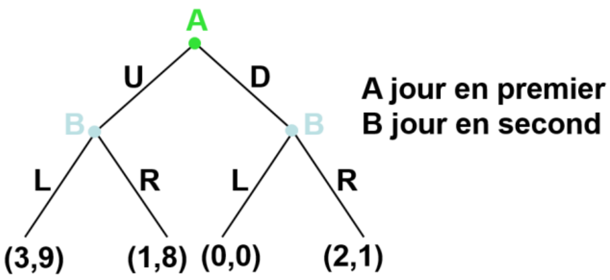
Pour illustrer ce concept revenons à l'exemple précèdent représenté comme suit :

		joueur B	
		L	R
joueur A	U	(3,9)	(1,8)
	D	(0,0)	(2,1)

(U,L) et **(D,R)** sont **deux équilibres de Nash** quand le jeu est simultané, il **est impossible** de savoir quel équilibre va arriver.

Supposons maintenant que le jeu est **séquentiel** : **A est le leader** et **B le follower**.

Nous pouvons réécrire ce jeu sous sa **forme extensive** :



A partir de cette représentation on peut déduire ce qui suit :

- **(U,L)** est un équilibre de Nash
- **(D,R)** est un équilibre de Nash

Question : Quel est donc celui qui va sortir sachant que **le jeu est séquentiel** ?

Réponse :

Tout dépond de **A** (Premier joueur). Une analyse des actons de A mène à :

- Si A joue **U** alors forcément B joue **L** (pour maximiser son gain) ; et **A** gagne **3**.
- Si A joue **D** alors forcément **B** joue R (pour maximiser son gain)); et **A** gagne **2**.

Etant donné que **A** est le premier qui va jouer il jouera **U (pour maximise son gain 3>2)**, et Donc **(U,L)** est l'équilibre de Nash qui sortira.

1.4.2 Exercice :

Faire la résolution de ce jeu séquentiel en inversant l'ordre des joueurs (B est le premier suivi de **A**). Qu'il est l'équilibre de Nash qui va sortir ?

2 Fonctions de meilleures réponses

La fonction de meilleure réponse (ou "best response function" en anglais) est un concept clé en théorie des jeux, en particulier lors de l'analyse des équilibres de Nash. La fonction de meilleure réponse représente les choix optimaux d'un joueur en réponse aux actions des autres joueurs.

Plus précisément, pour chaque joueur dans un jeu, la fonction de meilleure réponse indique quelles sont les meilleures stratégies disponibles pour ce joueur, compte tenu des stratégies choisies par les autres joueurs.

2.1 Formulation

La fonction de meilleure réponse du joueur $\boldsymbol{i}$ est la fonction $\boldsymbol{B_i}$ qui associe à chaque combinaison de stratégies des autres joueurs $\boldsymbol{s_{-i}}$ les stratégies du joueur $\boldsymbol{i}$ qui maximise son utilité : $\boldsymbol{B_i}(\boldsymbol{s_{-i}})$

$$= \{ \boldsymbol{s_i} \in S_i \text{ t.q. } \boldsymbol{\Pi}^{i}(\boldsymbol{s_i}, \boldsymbol{s_{-i}}) \geq \boldsymbol{\Pi}^{i}(\boldsymbol{s_i^J}, \boldsymbol{s_{-i}}) \text{ pour tout } \boldsymbol{s_i^J} \in \boldsymbol{S_i} \}$$

Un équilibre de Nash est un profil **s*** tel que la stratégie du joueur $\boldsymbol{i}$ est une meilleure réponse :

$$\boldsymbol{S^*_i} \in \boldsymbol{B_i}(\boldsymbol{s^*_{-i}}) \text{ pour tout } i \in N$$

2.2 Illustration à l'aide d'exemple

Soit un jeu **2×2**; *i.e.*, un jeu avec deux joueurs **A** et **B**, qui ont chacun deux actions possibles

A \ B	a^B_1	a^B_2
a^A_1	(6 ,4)	(3 , 5)
a^A_2	(4 , 3)	(5, 7)

- **A** peut choisir entre deux actions : $\mathbf{a^A_1}$ et $\mathbf{a^A_2}$
- **B** peut choisir entre deux actions $\mathbf{a^B_1}$ et $\mathbf{a^B_2}$
- Il y a 4 paires d'action possibles $(\mathbf{a^A_1}, \mathbf{a^B_1}), (\mathbf{a^A_1}, \mathbf{a^B_2}), (\mathbf{a^A_2}, \mathbf{a^B_1}), (\mathbf{a^A_2}, \mathbf{a^B_2})$
- Chaque **paire** d'action donnera des gains différents aux joueurs

A Partir de la représentation de jeu, les gains des joueurs A et B quand ils choisissent respectivement les actions a^A_1 et a^B_1 sont :

$$\boldsymbol{\Pi}^A(a^A_1, a^B_1) = 6 \text{ et } \boldsymbol{\Pi}^B(a^A_1, a^B_1) = 4$$

$\Pi^A(a^A_1, a^B_2) = 3$ et $\Pi^B(a^A_1, a^B_2) = 5$

$\Pi^A(a^A_2, a^B_1) = 4$ et $\Pi^B(a^A_2, a^B_1) = 3$

$\Pi^A(a^A_2, a^B_2) = 5$ et $\Pi^B(a^A_2, a^B_2) = 7$

Pour déterminer **l'équilibre de Nash par la fonction de meilleure réponse,** on doit déterminer la meilleure réponse de **A** compte tenu des actions de **B**, puis la meilleure réponse de **B** compte tenu des actions de **A.** Pour se faire on doit répondre aux questions suivantes :

a- Fonction de la meilleure réponse pour le joueur A

- **Question 1** : Si **B** choisit l'action a^B_1, quelle est la meilleure réponse de **A** ?

Réponse 1 :

La meilleure réponse de **A** est a^A_1 car $\Pi^A(a^A_1, a^B_1) = 6 > \Pi^A(a^A_2, a^B_1) = 4$

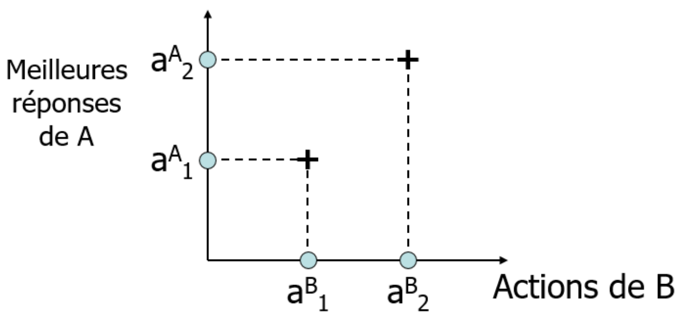
- **Question 2** : Si **B** choisit l'action a^B_2, quelle est la meilleure réponse de **A** ?

Réponse 2 :

La meilleure réponse de **A** est a^A_2 car $\Pi^A(a^A_2, a^B_2) = 5 > \Pi^A(a^A_1, a^B_2) = 3$

- Si **B** choisit a^B_1 alors A choisit a^A_1
- Si **B** choisit a^B_2 alors A choisit a^A_2

- La **"courbe" de meilleure réponse** de A est donc :



b- Fonction de la meilleure réponse pour le joueur B

- **Question 3** : Si **A** choisit l'action a^A_1, quelle est la meilleure réponse de **B** ?

Réponse 3 :

La meilleure réponse de **B** est a^B_2 car $\Pi^B(a^A_1, a^B_1) = 4 < \Pi^B(a^A_1, a^B_2) = 5$

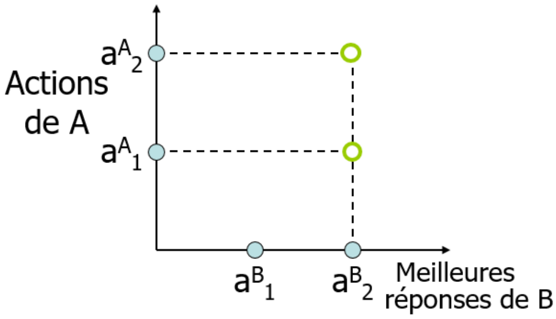
- **Question 4** : Si **A** choisit l'action a^A_2, quelle est la meilleure réponse de **B** ?

Réponse 4 :

la meilleure réponse de **B** est a^B_2 car $\Pi^B(a^A_2, a^B_1) = 3 < \Pi^B(a^A_2, a^B_2) = 7$

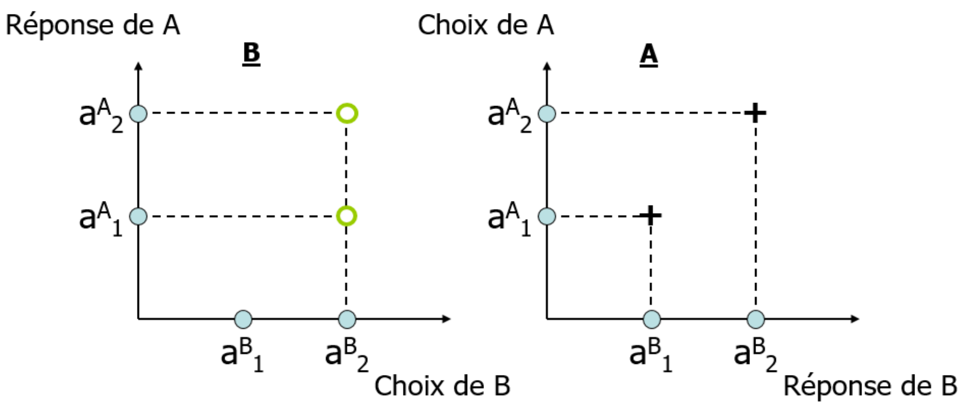
- Si **A** choisit $\mathbf{a}^{A}_{1}$ alors A choisit $\mathbf{a}^{B}_{2}$
- Si **A** choisit $\mathbf{a}^{A}_{2}$ alors A choisit $\mathbf{a}^{B}_{2}$

- La **“courbe” de meilleure réponse** de **B** est donc :

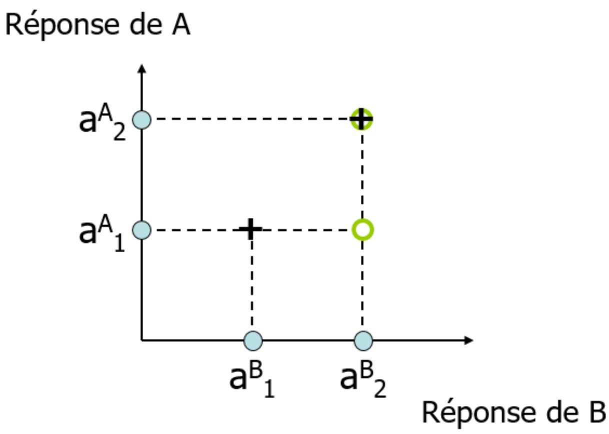


Comment peut-on utiliser les courbes de meilleures réponses pour localiser les équilibres **de Nash** du jeu ?

=> **Superposez les courbes…**



Une fois les deux courbes sont superposées on aura la figure ci-dessous :



Dans cette figure le point d’intersection des courbe est $\mathbf{a}^{A}_{2}$ et $\mathbf{a}^{B}_{2}$, qui constitue l’équilibre de Nash,

d’où le profil **($\mathbf{a}^{A}_{2}$, $\mathbf{a}^{B}_{2}$) est un équilibre de Nash** de ce jeu

Remarque :

- $\mathbf{a}_2^A$ est une meilleure réponse à $\mathbf{a}_2^B$
- $\mathbf{a}_2^B$ est une meilleure réponse à $\mathbf{a}_2^A$
-

2.2.1 *Exercice*

- **Enoncé**

Soit le jeu suivant sous la forme normale suivante

G1 \ GP2	U	V
X	4,2	6,5
Y	8,3	4,4

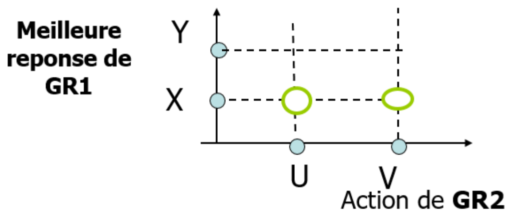
Trouvez son équilibre de Nash par la fonction de meilleure réponse

- **Solution**

a- Traçons la **Courbe de meilleure réponse GR1**

- Si **GP2** choisit **U** alors **GP1** choisit **X**
- Si **GP2** choisit **V** alors **GP1** choisit **X**

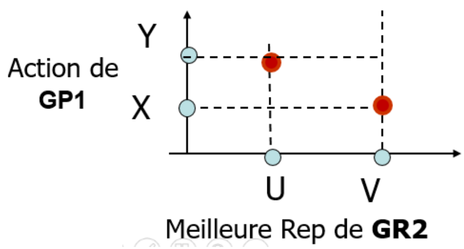
D'où la courbe Meilleure réponse de **GR1 est**



b- Traçons la **Courbe de meilleure réponse GR2**

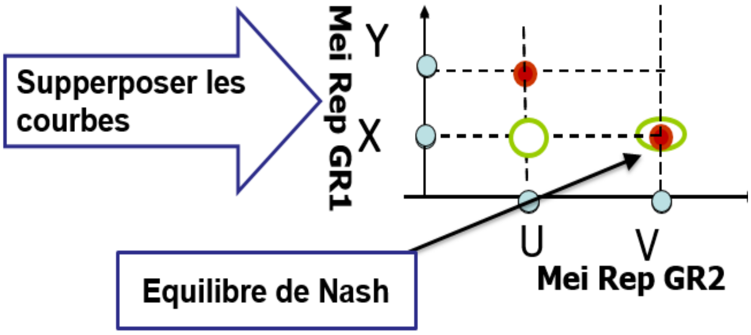
- Si **GP1** choisit **X** alors **GP2** choisit **V**
- Si **GP1** choisit **Y** alors **GP2** choisit **U**

D'où la courbe Meilleure réponse de **GR2 est**



c- La superposition **des deux courbes :**

La superposition n comme le montre la figure ci-dessous, **nous** donne le profil **(X,V)** comme intersection, d'où **(X,V)** est l'équilibre de Nash de ce jeu,



3 Equilibre de Nash en Stratégie Mixte :

L'équilibre de Nash en Stratégie Mixte est utilisé pour analyser les jeux dans lesquels les joueurs prennent des décisions stratégiques tout en étant confrontés à une certaine incertitude concernant les actions de leurs adversaires. Il est particulièrement utile pour modéliser des situations où les joueurs ne choisissent pas une stratégie pure (c'est-à-dire une stratégie unique), mais plutôt une distribution de probabilités sur un ensemble de stratégies possibles.

Voici une introduction aux principaux concepts liés à l'équilibre de Nash en stratégie mixte :

1. Stratégies mixtes : Une stratégie mixte est une stratégie dans laquelle un joueur attribue des probabilités à ses différentes actions possibles. Par exemple, au lieu de choisir toujours la même action, un joueur peut décider de jouer une action avec une certaine probabilité et une autre action avec une probabilité différente.
2. Profil de stratégies mixtes : Dans un jeu à deux joueurs, un profil de stratégies mixtes consiste en une paire de stratégies mixtes, une pour chaque joueur. Cela représente comment chaque joueur distribue ses probabilités sur ses actions possibles.
3. Équilibre de Nash en stratégie mixte : Un équilibre de Nash en stratégie mixte est une situation dans laquelle chaque joueur, connaissant les stratégies mixtes de ses adversaires, n'a aucune incitation à modifier sa propre stratégie mixte. En d'autres termes, chaque joueur a atteint une situation où il maximise son utilité (ou son gain) compte tenu des choix probabilistes de ses adversaires.
4. Résolution de jeux : L'équilibre de Nash en stratégie mixte permet de résoudre de nombreux jeux où les joueurs ont des informations imparfaites sur les actions de leurs adversaires, ou lorsqu'il existe des actions dominées (actions qui sont toujours pires que d'autres). Il est souvent utilisé pour analyser des jeux tels que le dilemme du prisonnier, le jeu de pierre-papier-ciseaux, ou d'autres jeux de hasard.

En résumé, l'équilibre de Nash en stratégie mixte est un concept clé de la théorie des jeux qui permet de modéliser des situations où les joueurs prennent des décisions sous forme de probabilités. Il s'agit d'une solution au jeu dans laquelle aucun joueur n'a intérêt à changer sa stratégie compte tenu des choix probabilistes de ses adversaires. Cette notion est essentielle pour comprendre et analyser une grande variété de situations de prise de décision stratégique.

3.1 Petit Rappel

- Une **stratégie pure** du joueur i est un plan d'action qui prescrit une action de ce joueur pour chaque fois qu'il est susceptible de jouer. On note par S_i l'ensemble des stratégies **pures** du joueur i et par s_i une stratégie **pure** de ce joueur.
- Une **stratégie mixte** du joueur i est une distribution de probabilités p_i définie sur l'ensemble des stratégies pures du joueur i.

  On note Σ_i l'ensemble des **stratégies mixtes** du joueur i et par σ_j **une stratégie mixte** de ce joueur i ➔ $\Sigma_i = \{\sigma_1, \sigma_2 \ \ldots.. \ \sigma_q\}$

3.2 Définition

Un équilibre de **Nash en stratégies mixtes** est un profil de stratégies mixtes $\sigma^* = (\sigma^*_i, \sigma^*_{-i}) \in \Sigma$ tel que pour tout i et tout $\sigma_i \in \Sigma_i$ on a

$$\Pi^i(\sigma^*_i, \sigma^*_{-i}) \geq \Pi^i(\sigma_i, \sigma^*_{-i})$$

3.3 Théorème.

σ^* est un équilibre de Nash si et seulement si pour tout i et tout $s_i \in S_i$

$$\Pi^i(\sigma^*_i, \sigma^*_{-i}) \geq \Pi^i(s_i, \sigma^*_{-i})$$

***Théorème. [Nash, 1950] :** Tout jeu sous forme stratégique a un équilibre de Nash en stratégies mixtes.

3.4 La résolution d'un jeu en équilibre de Nash en stratégie

La résolution d'un jeu en équilibre de Nash en stratégie mixte implique généralement de suivre une série d'étapes pour trouver les probabilités auxquelles chaque joueur devrait jouer pour atteindre cet équilibre. Voici un algorithme de base pour résoudre un jeu en équilibre de Nash en stratégie mixte :

1. **Définition du jeu** : Tout d'abord, identifiez le jeu en question, y compris les joueurs, les stratégies possibles pour chaque joueur, les gains associés à chaque combinaison de stratégies, et les informations dont les joueurs disposent.
2. **Formalisation des stratégies mixtes** : Représentez les stratégies mixtes de chaque joueur sous forme de distributions de probabilités. Chaque joueur attribue des probabilités à ses différentes

stratégies. Par exemple, dans un jeu à deux joueurs, un joueur peut jouer A avec une probabilité p et B avec une probabilité 1-p.

3. **Écriture des fonctions de gains** : Pour chaque joueur, écrivez une fonction de gain qui mesure son utilité en fonction des probabilités choisies par lui et son adversaire. Les gains sont basés sur les résultats de chaque combinaison de stratégies.
4. **Recherche d'équilibre de Nash** : Pour résoudre le jeu en équilibre de Nash en stratégie mixte, suivez les étapes suivantes :

a. Pour chaque joueur, maximisez sa fonction de gain en ajustant ses probabilités. Cela revient à trouver les valeurs des probabilités qui maximisent son utilité tout en supposant que l'autre joueur joue selon une certaine distribution de probabilités.

b. Répétez cette étape pour chaque joueur, en considérant les probabilités du joueur adverse telles qu'elles ont été trouvées à l'étape précédente.

c. Continuez à ajuster les probabilités jusqu'à ce que vous trouviez un point où aucune stratégie ne donne à un joueur une incitation à changer sa distribution de probabilités, c'est-à-dire que vous avez trouvé un équilibre de Nash en stratégie mixte.

5. **Vérification de l'équilibre** : Une fois que vous avez trouvé un ensemble de probabilités pour chaque joueur, assurez-vous que personne n'a intérêt à changer sa stratégie en fonction des probabilités de l'autre joueur. C'est l'équilibre de Nash en stratégie mixte.
6. **Présentation des résultats** : Présentez les probabilités auxquelles chaque joueur devrait jouer pour atteindre l'équilibre de Nash en stratégie mixte, ainsi que les gains résultants pour chaque joueur à cet équilibre.

Il est important de noter que la résolution de jeux en équilibre de Nash en stratégie mixte peut être complexe, en particulier pour les jeux à de nombreux joueurs ou les jeux avec des espaces de stratégies étendus. Dans de tels cas, des outils mathématiques plus avancés, tels que la programmation linéaire ou la théorie de la décision, peuvent être nécessaires pour trouver les solutions.

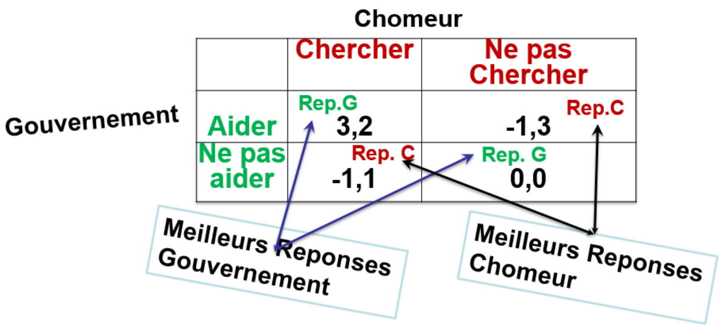
3.5 Exemple Haut du formulaire

La matrice ci-dessous présente le jeu en stratégie pure (**Chercher ou non --- aider ou non**)

		Chomeur	
		Chercher	Ne pas Chercher
Gouvernement	Aider	3,2	-1,3
	Ne pas aider	-1,1	0,0

Le gouvernement évite aider les chômeurs qui ne cherchent pas ou encore n'aide pas les chômeurs qui cherchent

Interaction stratégique : **Si on regarde les meilleures réponses**



On voit bien que ce jeu ce jeu n'a pas **d'équilibre en stratégie pure,** car à **peu importe le profil** toujours un des joueurs a intérêt de changer sa stratégie pour améliorer ces gains.

- **Si on passe à une stratégie mixe**

On passe à des comportements qui ne sont pas ferme aider un peu ou beaucoup, chercher des fois des fois non etc …autrement dit on associe à chaque action une probabilité. Comme ça il serait possible d'avoir un équilibre à ce jeu en stratégie mixte.

Ça consiste à supposer que le chômeur cherche avec une probabilité **α** comprise entre 0 et 1

Et le gouvernement aide avec une probabilité **β** (entre 0 et 1) (par exemple aider en fonction du son budget (30 % ou **40%** …).

- La valeur de **α** détermine la stratégie **mixte** de Chômeur
- La valeur de **β** détermine la stratégie **mixte** de Gouvernement

On va associer **α** pour l'action « **Chercher** » et **1- α Pour** l'action « **Ne pas chercher ».**

On associe aussi la probabilité **β** pour l'action « **aider** » et **1- β** pour l'action « **Ne pas aider ».**

On a mis ces probabilités pour trouver l'Esperance mathématique des joueurs pour voir comment ils se comportent.

Et la représentation de jeu en stratégie mixte devient comme suit :

Chômeur

Gouvernement	Chercher Prob α	Ne pas Chercher Prob (1-α)
Aider Prob β	3,2	-1,3
Ne pas aider Prob 1- β	-1,1	0,0

Comment trouver l'équilibre à partir de la meilleure solution et :

- Dire que le chômeur va chercher avec **la Proba α** et le Gvrnm aide avec **la Proba β**.

- Et surtout donner les valeurs de **α** et **β** pour

a. Calcul de l'Esperance Mathématique du Chômeur

Le calcul de l'Esperance mathématique du **chômeur se fait en fonction de ses gains et de la probabilité β** et (1- β) associées aux actions du **Gouvernement**. Ainsi on aura :

Esp(Π *Chomeur* (Cherch))= 2 β+1*(1- β)= β +1

Esp(Π *chomeur* (Pas cherch)= 3β +0*(1- β)= 3β

La figure ci-dessous montre et explique comment on a effectué les opérations de calcul.

Chomeur

Gouvernement	**Chercher** Prob α	**Ne pas Chercher** Prob (1-α)
Aider Prob β	3 , 2	-1 , 3
Ne pas aider Prob 1- β	-1 , 1	0 , 0

Espérance Math du Chômeur

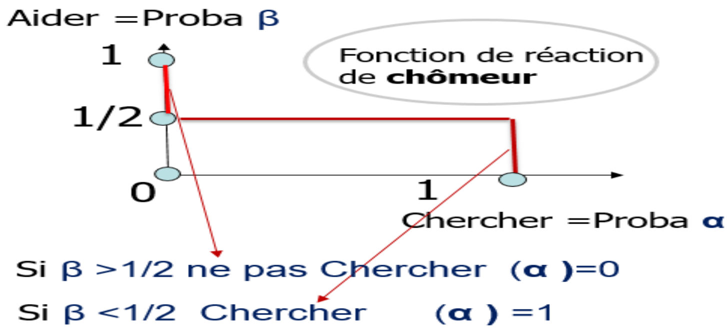
➔ **Esp(Π *Chomeur* (Cherch))**= 2 β+1*(1- β)= β +1

Esp(Π *chomeur* (Pas cherch)= 3β +0*(1- β)= 3β

Ce résultat sera interprété comme suit :

- Quand **3β > β +1** ➔ **β >1/2** ➔ le **chômeur** choisira « **ne Pas chercher** »
- Quand **3β < β +1** ➔ **β <1/2** ➔ le **chômeur** choisira « **de chercher** »
- Valeur **β** pour l'équilibre **β=1/2**
- **La fonction de réaction de Chômeur:**La meilleure réponse du **Chômeur** en fonction **de β**
  - **Si** β<1/2 Alors **α =1** autrement Si le **Gvrnmt n' aide pas,** le **chômeur** est obligé de **chercher**
  - **Si β>1/2** Alors **α =0 autrement** Si le **Gvrnmt aide,** le **chômeur ne cherche pas**
  - **Si** β **=1/2** Alors **α** peut prendre toutes les valeurs, le **chômeur est indifférent**

La fonction de meilleures réponses de Chômeur est comme suit :



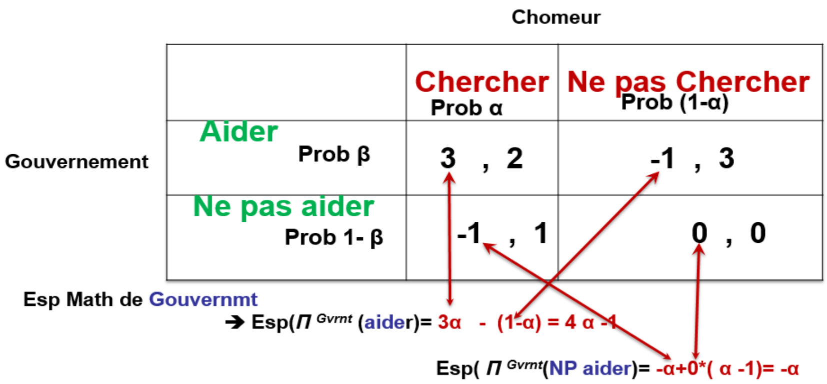
b. Calcul de l'Esperance Mathématique du Gouvernement

Le calcul de l'Esperance mathématique du **Gouvernement se** fait en fonction de **ses gains** et de la **probabilité α** et (1- **α**) associées aux actions du **Chômeur**. Ainsi on aura :

- Esp(Π^{Gvrnt} (aider))= 3α - (1-α) = 4 α -1

- Esp(Π^{Gvrnt} (Ne Pas aider)= -α+0*(α -1)= -α

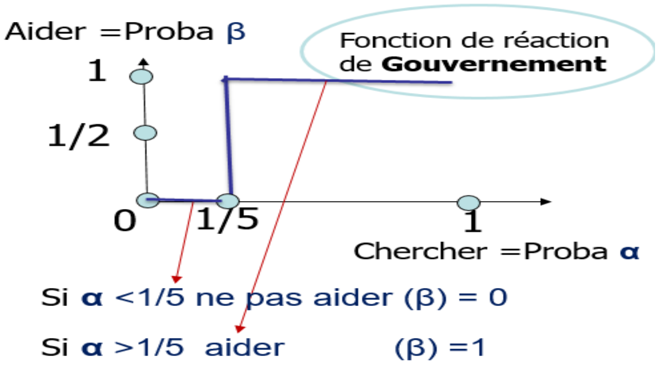
La figure ci-dessous montre et explique comment on a effectué les opérations de calcul.



Ce résultat sera interprété comme suit :

- Quand ➔ 4α -1 < -α (ie **α <1/5)** ➔ **le Gouvernement choisira** « aider »
- Quand ➔ 4α -1 > -α (ie **α >1/5)** ➔ **le Gouvernement choisira de « ne pas** aider »
- Valeur **α** pour l'équilibre **α** = **1/5**
- **La fonction de réaction de Gouvernement :** Les meilleures réponses du **Gouvernement** en fonction de **α**
  - **Si α<1/5** Alors **β=0,** le **chômeur** « ne cherche pas » donc le **Gouvernement** « **n'aide pas** »
  - **Si α >1/5** Alors **β=1,** le **chômeur** « **cherche »** le **Gouvernement** « **aide »**
  - **Si α =1/5** Alors **β** peut prendre toutes les valeurs, le **Gouvernement** **est indifférent**

L'allure de La fonction de meilleures réponses de Chômeur est comme suit :



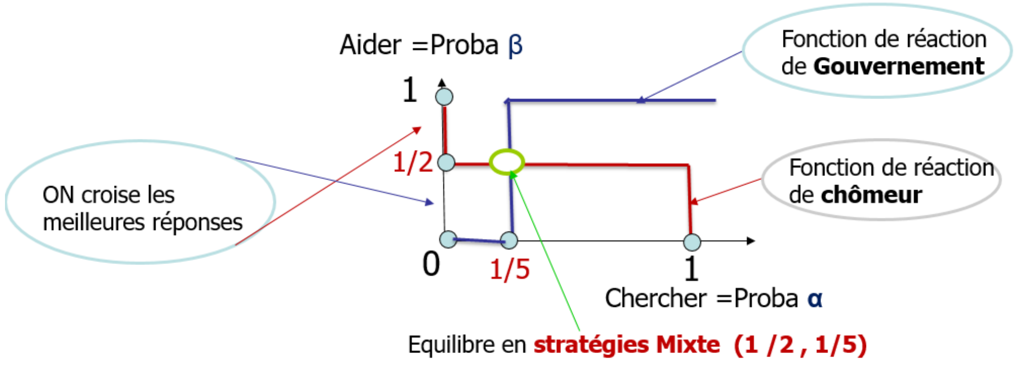
Σ = { Σ Chom , Σ Gvnt } d'où

- Σ_{chom} = (σ_{11} , σ_{12}) = (1/5 *chercher , 4/5 * ne pas chercher)
- Σ_{Gvnt} = (σ_{21} , σ_{22}) = (1/2 aider , 1/2 * ne pas aider)

Remplaçant les probabilités α **et β** par les valeurs trouvées la représentation de jeu en stratégie mix devient

		Chomeur	
		Chercher $\alpha = 1/5$	Ne pas Chercher $1-\alpha=4/5$
Gvrnmt	Aider $\beta = 1/2$	3 , 2	-1 , 3
	Ne pas aider $1-\beta=1/2$	-1 , 1	0 , 0

- Pour déterminer l'équilibre en stratégie Mixte résultant on Croise les deux courbe trouvés précédemment, comme le montre la figure ci-dessous



Donc il y a une seule valeur pour laquelle il y a une double meilleure réponse c'est ***β =1/2*** et ***α=1/5*** là où se croisent les courbes nous donne **l'équilibre** en **Stratégie mixte**

c) Interprétation et lecture de comportement des joueurs dans ce jeu

Dans ce jeu le **chômeur** va chercher 1 fois / 5 (**α=1/5)** et le **Gouvernement** va aider 1 fois / 2 (**β =1/2)**

Une fois on a trouvé les α et β ce qui est intéressant, on va calculer pour ces deux joueurs la valeur de jeu dans le cadre d'un équilibre en stratégie mixte : il suffit de reprendre les espérances mathématiques des deux joueurs :

- **Esp-chom= $2\alpha\beta + 1*\alpha(1-\beta) + 3(1-\alpha)\beta + 0(1-\beta)(1-\alpha) = 1,5$**
- **Et Esp-Gvrn = $3\alpha\beta - 1*\alpha(1-\beta) - 1(1-\alpha)\beta + 0(1-\beta)(1-\alpha) = -1/5$**

En jouant ce jeu le **chômeur** espère gagner **1,5** et le **Gouvernement** **-1/5**. Par ces valeurs le **chômeur** et le **Gouvernement** auraient pu éviter des valeur **catastrophiques** (Dégâts) **0** pour le **chômeur** et **– 1** pour le **Gouvernement.**

3.5.1 Exercice

Considérons un nouveau jeu...

		joueur B	
		L	R
joueur A	U	(1,2)	(0,4)
	D	(0,5)	(3,2)

1) Existe-t-il un équilibre de Nash en stratégie pure ?

2) Trouver un équilibre en stratégie mixte

En précisant la valeur du jeux en stratégie mixte pour chaque joueur

Comparer chaque valeur avec celle du pire des cas en stratégie pure

Réponse :

1) Non

2) Esperance de A = 2*Pa*Pb + 4*(1-Pa)*Pb + 5*Pa*(1-Pb) + 2*(1-Pa)*(1-Pb)

Esperance de B = 1*Pa*Pb + 0*(1-Pa)*Pb + 0*Pa*(1-Pb) + 3*(1-Pa)*(1-Pb)

Simplifier EspA et EspB puis calculer Pb et Pa telle que les dérivées d**(EspA)/d(Pa)=0** et **d(EspB)/d(Pa)=0**

Proba : Pb=**3/5** et proba Pa=**3/4**

val Esperance de A = 2*3/4*3/5 + 4*1/4*3/5 + 5*3/4*2/5 + 2*1/4*2/5

val Esperance de B = 1*3/4*3/5 + 0*1/4*3/5 + 0*3/4*2/5 + 3*1/4*2/5