IFT339
Structures de données

Thème 14 :Graphes

Aïda Ouangraoua
Département d'informatique
UNIVERSITÉ DE SHERBROOKE

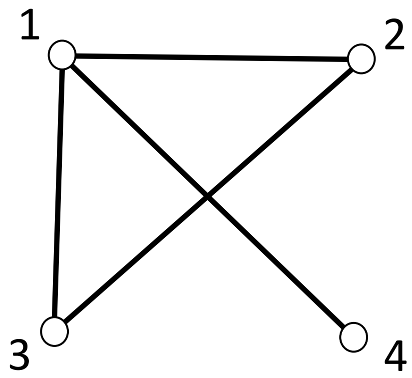
Graphes

- **Exemples**:
  - Lignes aériennes (aéroports, vols)
  - Réseaux routiers (villes, routes)
  - Web (pages, liens hypertextes)
  - Réseaux sociaux (personnes, amitiés)

Graphes

- **Définition:**
  - **Paire G=(V,E)** avec V ensemble de sommets et E ensemble d'arêtes représentées par des paires de sommets

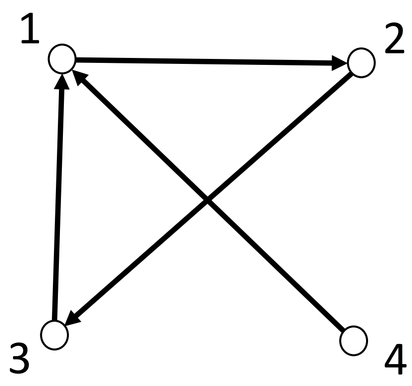
V = {1,2,3,4} ; E = {(1,2), (1,3), (1,4),(2,3)}



Graphe orienté vs non-orienté

- **Graphe orienté** : les arêtes (arcs) ont une direction (relations entre sommets non-symétriques)

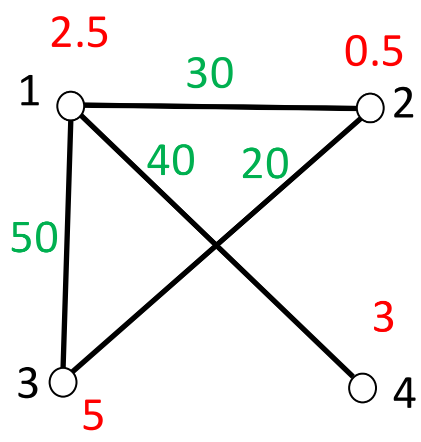
V = {1,2,3,4} ; E = {(1,2), (3,1), (4,1),(2,3)}



Graphe à sommets ou arêtes pondérés

- **Graphe à sommets pondérés** : fonction poids définie de V → ℝ
- **Graphe à arêtes pondérées** : fonction poids définie de E → ℝ

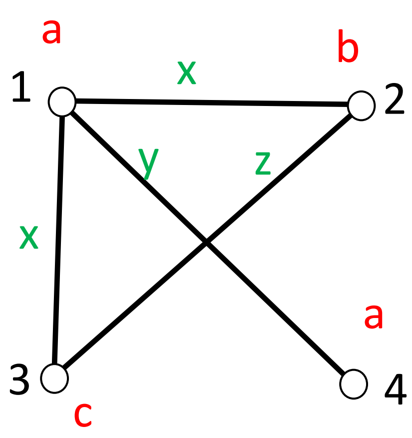
V = {1,2,3,4} ; E = {(1,2), (1,3), (1,4),(2,3)}



Graphe à sommets ou arêtes étiquetés

- **Graphe à sommets étiquetés** : fonction définie de V→ Σ_sommets
- **Graphe à arêtes étiquetées** : fonction définie de E → Σ_aretes
- Σ_sommets, Σ_aretes : alphabets

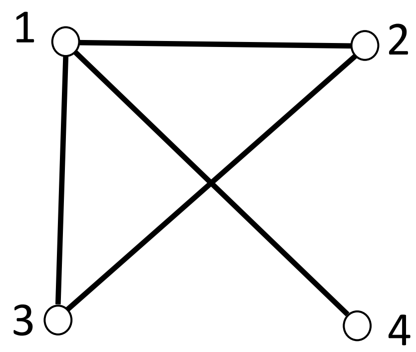
V = {1,2,3,4} ; E = {(1,2), (1,3), (1,4),(2,3)}



Graphes

- **Ordre du graphe** : le nombre de ses sommest
- **Sommets adjacents ou voisins** : reliés par une arête (arc)
- **Degré d'un sommet** : nombre d'arêtes reliés au sommet

V = {1,2,3,4} ; E = {(1,2), (1,3), (1,4),(2,3)}



Ordre : 4
Voisin de 1 : {2,3,4}
Degré de 1 : 3

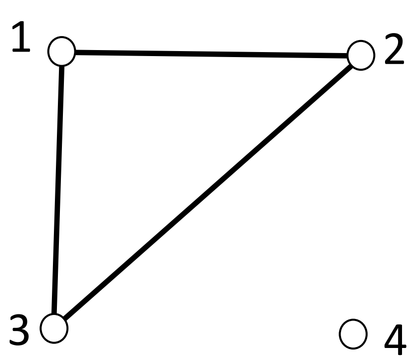
Graphes

- **Chaîne de longueur n** : suite de n arêtes qui relie un sommet à un autre ou lui-même

- **Cycle** : chaîne qui relie un sommet à lui-même en passant au plus une fois par chaque sommet

- **Graphe connexe** : il existe une chaîne de longueur n >= 1entre toute paire de sommets

V = {1,2,3,4} ; E = {(1,2), (1,3),(2,3)}



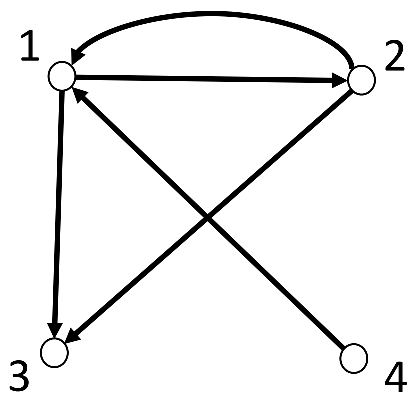
Chaîne de longueur 2 : 1 2 3
Cycle de longueur 3 : 1 2 3 1
Graphe non-connexe

Graphes

- **Chemin** : chaîne bien orientée

- **Circuit** : cycle bien orienté

V = {1,2,3,4} ; E = {(1,2), (2,1), (1,3), (4,1),(2,3)}



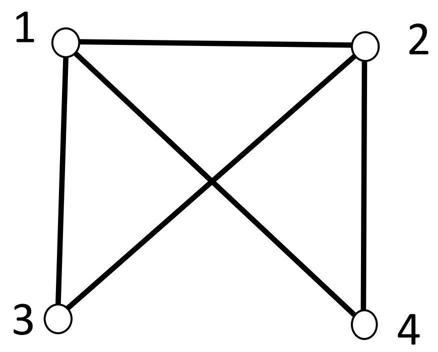
Chemin de longueur 2 : 1 2 3
Circuit de longueur 2 : 1 2 1

1 2 3 1 est un cycle, mais pas un circuit

Graphes

- **Chaîne hamiltonienne** : passant une seule fois par tous les sommets du graphe
- **Chaîne eulérienne** : passant une seule fois par toutes les arêtes du graphe

V = {1,2,3,4} ; E = {(1,2), (1,3), (1,4), (2,3), (2,4)}



Chaîne hamiltonienne: 3 1 2 4 ; 3 1 4 2

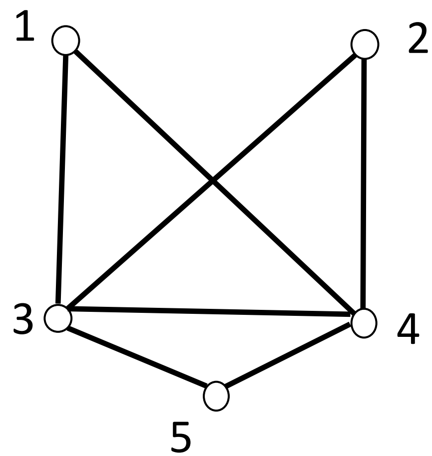
Chaîne eulérienne : 1 3 2 1 4 2

Graphes

- **Cycle hamiltonien** : passant une seule fois par tous les sommets du graphe

- **Cycle eulérien** : passant une seule fois par toutes les arêtes du graphe

V = {1,2,3,4} ; E = {(1,3), (1,4), (2,3), (2,4),(3,4), (3,5), (5,4)}



Cycle eulérien : 1 3 2 4 5 3 4 1

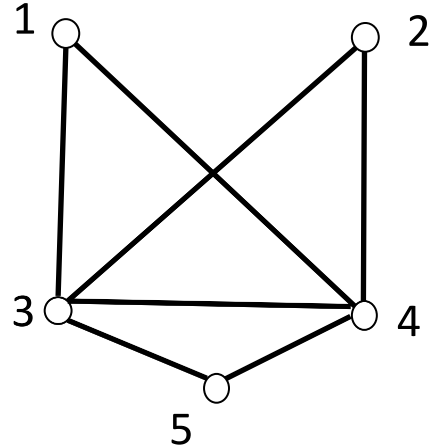
- **Graphe hamiltonien** : possède au moins un cycle hamiltonien

- **Graphe eulérien** : possède au moins un cycle eulérien

Graphe eulérien

- **Théorème d'Euler** : un graphe est eulérien si tous les sommets du graphe ont un degré pair.

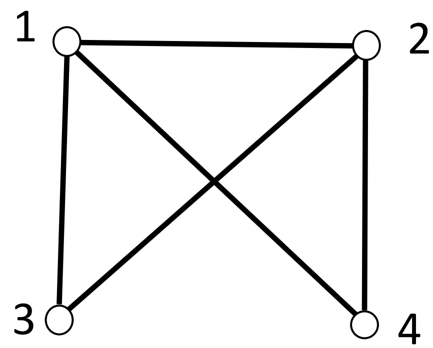
V = {1,2,3,4} ; E = {(1,3), (1,4), (2,3), (2,4),(3,4), (3,5), (5,4)}



Cycle eulérien : 1 3 2 4 5 3 4 1

- **Exercice 1** : donner un algorithme pour trouver un cycle eulérien pour un graphe donné G.

Représentation



- Ensemble de sommets + Ensemble de paires de sommets
  G = ({1,2,3,4}, {(1,2), (1,3), (1,4), (2,3), (2,4)})

- Classe Sommet avec attributs Sommet* Voisins, tableau de pointeurs pointant vers les sommets voisins.

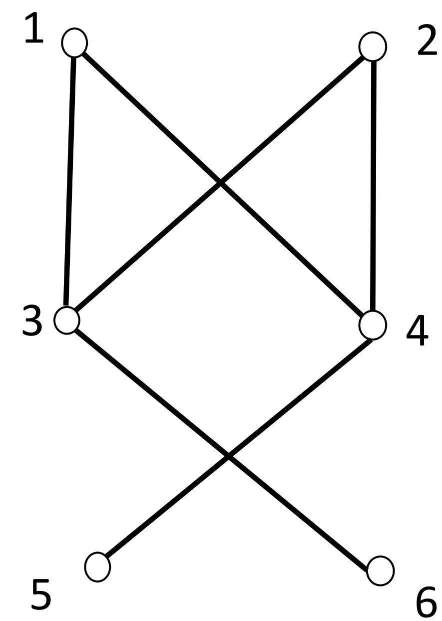
- Matrice d’adjacence : matrice binaire n x n, tel que n est le nombre de sommets

0	1	1	1
1	0	1	1
1	1	0	0
1	1	0	0

Parcours

- Pour trouver le chemin le plus court entre deux sommets

- **Parcours en largeur:**
  0- Choisir un sommet de départ S
  1- Lister S
  2- Lister tous les voisins de S
  3- Répéter 2 pour chaque voisin



Exemple: 1 3 4 2 6 5

- **Parcours en profondeur**
  0- Choisir un sommet de départ S
  1- Lister S (préfixe)
  2- Pour chaque voisin de S non listé, appliquer récursivement l'algorithme
  3- Si tous les voisins sont listés:
    - Lister S (postfixe)
    - revenir au sommet parent et continuer avec d'autres voisins

Exemple: 1 3 2 4 5 6 (prefixe) ; 6 5 4 2 3 1 (postfixe) ;

Parcours

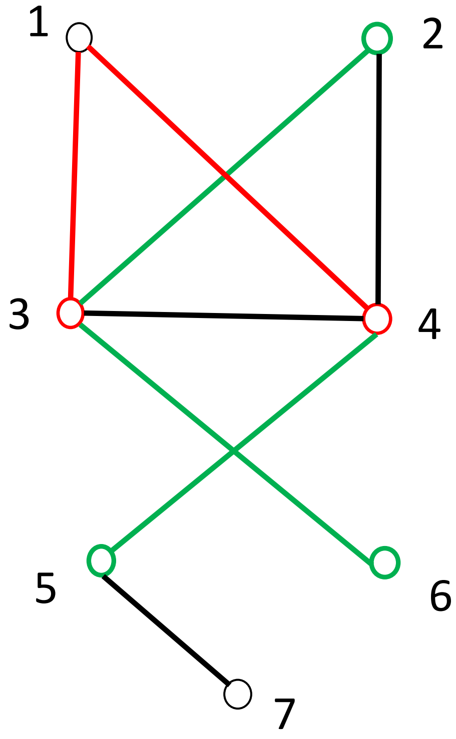
- Exercice 2 : Écrire l'algorithme du parcours en largeur pour un graphe représenté par une matrice d'adjacence M.
- Exercice 3 : Écrire l'algorithme du parcours en profondeur préfixe pour un graphe représenté par une matrice d'adjacence M.

M.size() : retourne le nombre de sommets du graphe
M[i][j] : accède à la valeur à la ligne i, colonne j

```
vector<int> parcours_largeur(M){




}
vector<int> parcours_profondeur_prefixe(M){




}
```

Parcours

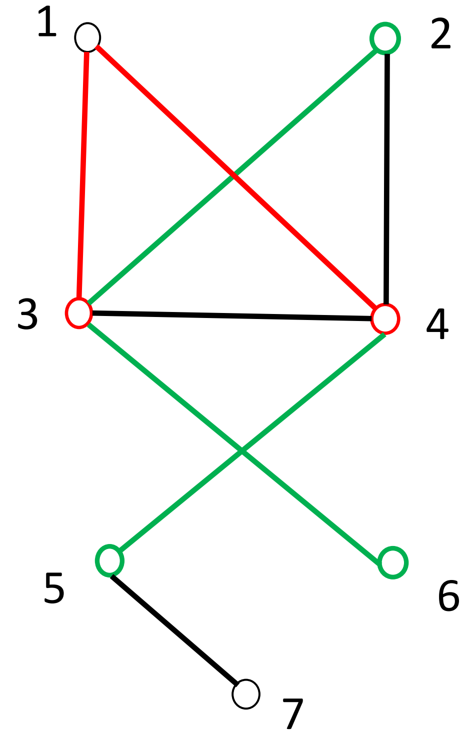
- Exercice 4 : Écrire un algorithme pour trouver un chemin optimal entre deux sommets i et j dans un graphe à arêtes non-pondérées (poids: 1).



Parcours

- Exercice 4 : Écrire un algorithme pour trouver un chemin optimal entre deux sommets i et j dans un graphe à arêtes non-pondérées (poids: 1).

- À partir de i:
- Trouver les sommets à distance 1
- Trouver les sommets à distance 2
- Continuer jusqu'à trouver j



Algorithme de Dijskstra

- Pour trouver un chemin optimal entre deux sommets i et j dans un graphe à arêtes pondérées

1- Affecter un poids provisoire 0 au sommet de départ i et attribuer provisoirement un poids ∞ aux autres sommets.

Répéter les étapes 2 et 3 tant que le sommet d'arrivée j n'a pas un poids définitif

2- Parmi les sommets dont le poids est provisoire, choisir un sommet x de poids minimal.
Affecter définitivement à x son poids

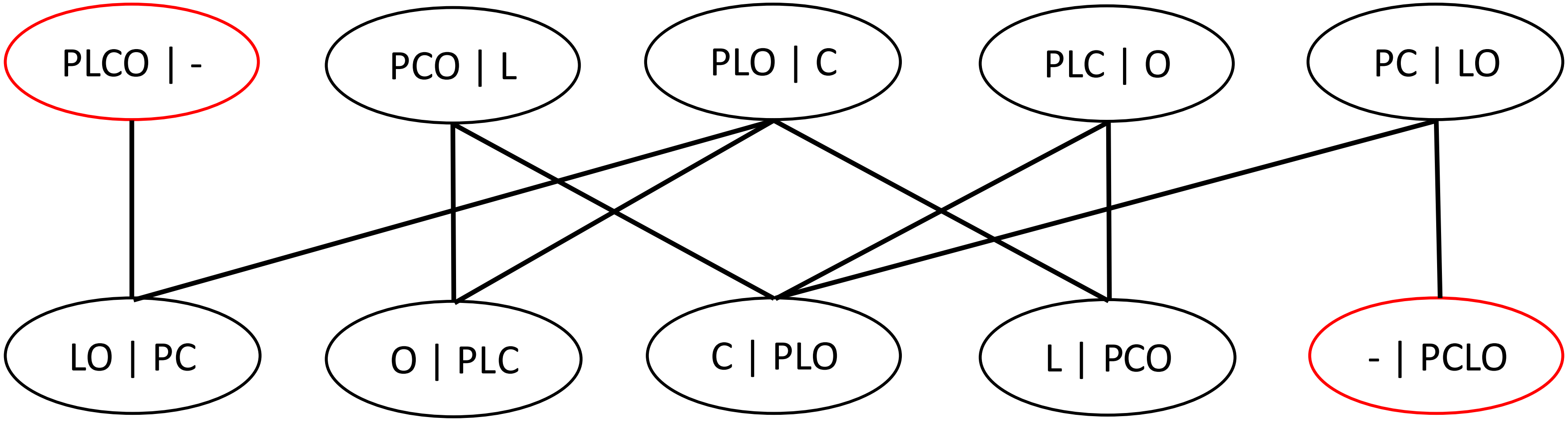
3- Pour tous les sommets adjacents à x qui n'ont pas de poids définitif, calculer leur poids en arrivant de x
Si le nouveau poids est inférieure au poids actuel, affecter provisoirement e nouveau poids et indiquer que l'ont vient de x pour ce poids.

Problème

- Un passeur (P) , un loup (L), une chèvre (C) et un chou (O) se trouvent sur la rive d'un fleuve. Le passeur doit les enmener sur l'autre rive mais, sa barque ne peut transporter qu'un seul d'entre eux à la fois. Comment doit-il procéder pour ne jamais laisser ensemble, en son absence, le loup et la chèvre, ou la chèvre et le chou ?

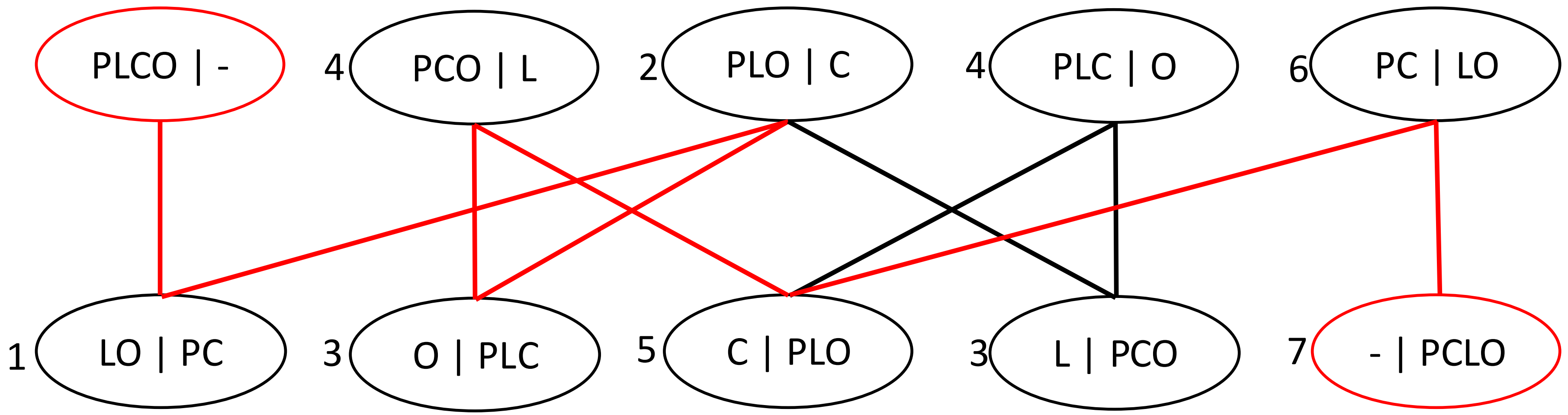
- Trouver un graphe et un problème sur ce graphe qui soit équivalent au problème ci-dessus.

Problème



- Les sommets du graphe sont les états possibles
- Il y a une arête entre deux sommets si on peut passer d'un état à l'autre avec un voyage
- Problème : on cherche un chemin (optimal) de (PLCO | -) à (- | PCLO)

Problème



- Les sommets du graphe sont les états possibles
- Il y a une arête entre deux sommets si on peut passer d'un état à l'autre avec un voyage
- Problème : on cherche un chemin (optimal) de (PLCO | -) à (- | PCLO)