L2 GÉNIE INFORMATIQUE • SEMESTRE 3

THÉORIE DES GRAPHES ET OPTIMISATION DES PROCESSUS

Séance 1 : Fondamentaux et Représentations

Ing. GBAGUIDI AISSE Alexandre

ESGC VERECHAGUINE A.K. • Cotonou, Bénin

Plan de la séance

🌍 Accroche

Les graphes dans votre quotidien

📚 Vocabulaire

Sommets, arêtes, degrés, voisinage

🔢 Types de graphes

Orientés, pondérés, simples

💾 Représentations

Liste vs matrice d'adjacence

✏️ Travaux dirigés

Exercices pratiques

💻 Travail personnel

Implémentation en C++

Objectifs d'apprentissage

À la fin de cette séance, vous serez capable de :

1. Définir formellement un graphe $G = (V, E)$
2. Calculer le degré d'un sommet et appliquer la propriété $\Sigma deg(v) = 2|E|$
3. Identifier le type d'un graphe (orienté, pondéré, simple, etc.)
4. Choisir la représentation appropriée (liste vs matrice)
5. Implémenter les structures de base en C++ (classes et méthodes)

Pourquoi étudier les graphes ?

🌐 Partout dans l'informatique

- Réseaux sociaux
- GPS et navigation
- Internet et routage
- Compilateurs

🔧 Outils essentiels

- Modélisation de problèmes
- Algorithmes fondamentaux
- Structures de données

💼 Employabilité

- Compétence attendue des ingénieurs
- Base pour l'IA et le Big Data
- Optimisation de systèmes

🚀 Innovation

- Recommandations (Netflix, YouTube)
- Détection de fraudes
- Analyse de réseaux

Modalités d'évaluation

12.5%	12.5%	20%	60%
Assiduité	**Contrôle Continu**	**Devoir**	**Projet**
Venir en cours ?	6 quiz de 10 min	Épreuve écrite 60 min	Code + Rapport + Soutenance

📋 **Détails importants**

- Quiz : Séances 2 à 7 (fin de séance)
- Devoir : Après 16h de cours
- Projet - Examen : En binôme

PARTIE 2

Les graphes

sont partout

Exemple 1 : Réseau social WhatsApp

Qui connaît qui dans votre promo ?

Chaque étudiant = sommet

Chaque contact = arête

> 🤔 **Questions intéressantes**
> • Qui est le plus connecté de la classe ?
> • Y a-t-il des groupes isolés ?
> • Combien de « sauts » pour atteindre quelqu'un ?

📱

Réseau social = Graphe non-orienté

Exemple 2 : GPS et Navigation

Trajet optimal Fidjrossè → Ganhi

Chaque carrefour = sommet

Chaque route = arête pondérée

Poids = temps de trajet ou distance

🚗 Applications

- Gozem : trajet le plus rapide
- Google Maps : éviter les embouteillages
- Livraison Jumia : optimiser les tournées

Exemple de graphe routier

Fidjrossè ↔ Cadjèhoun : 5 km

Cadjèhoun ↔ Étoile Rouge : 3 km

Étoile Rouge ↔ Ganhi : 4 km

Total : 12 km

Exemple 3 : Réseau Électrique SBEE

Distribution de l'énergie au Bénin

Chaque station = sommet

Chaque ligne électrique = arête

⚡ Problèmes à résoudre

- Pannes en cascade : si une station tombe,
  lesquelles sont affectées ?
- Connexion minimale : connecter tous les villages au moindre coût de câblage
- Fiabilité : identifier les stations critiques

💡 Optimisation réseau

Connecter 10 nouveaux villages avec le minimum de câble électrique

→ Algorithme des arbres couvrants minimaux (séance 5)

Exemples 4 & 5 : Compilation et Internet

💻 Compilation

Ordre des dépendances de modules

```
Module A → Module B
Module B → Module C
Module C → Module D
```

Problème : Dans quel ordre compiler ?

→ Tri topologique (séance 3)

🌐 Internet

Routage des paquets entre serveurs

- Routeurs = sommets
- Connexions = arêtes
- Latence = poids

Problème : Chemin le plus rapide ?

→ Dijkstra (séance 4)

Un outil mathématique universel

📱 **Réseaux sociaux**

🚗 **GPS**

💡 **Réseau électrique**

💻 **Compilation**

🌐 **Internet**

🎮 **Jeux vidéo**

Un même formalisme mathématique
pour résoudre des problèmes très différents !

PARTIE 3

Vocabulaire Fondamental

Les briques de base pour parler des graphes

Définition formelle d'un graphe

$$G = (V, E)$$

V = Vertices

Ensemble des sommets (ou nœuds)

Exemple : V = {A, B, C, D}

|V| = n (ordre du graphe)

E = Edges

Ensemble des arêtes (ou arcs)

Exemple : E = {(A,B), (B,C), (C,D)}

|E| = m (taille du graphe)

Sommet (Vertex)

Qu'est-ce qu'un sommet ?

Un sommet (ou nœud) est une entité du système modélisé.

🔷 **Exemples concrets**

- Réseau social : une personne
- Réseau routier : un carrefour, une ville
- Page web : une URL
- Compilation : un module de code

Un sommet

Arête (Edge)

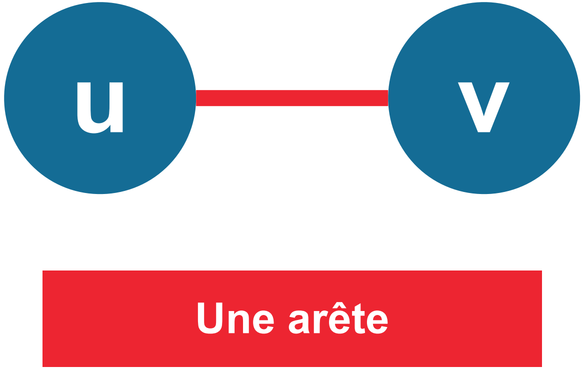
Qu'est-ce qu'une arête ?

Une arête (ou arc) est une connexion entre deux sommets.

Notation : e = (u, v)

🔷 Exemples concrets

- Réseau social : une amitié, un contact
- Réseau routier : une route entre deux villes
- Page web : un lien hypertexte
- Compilation : une dépendance



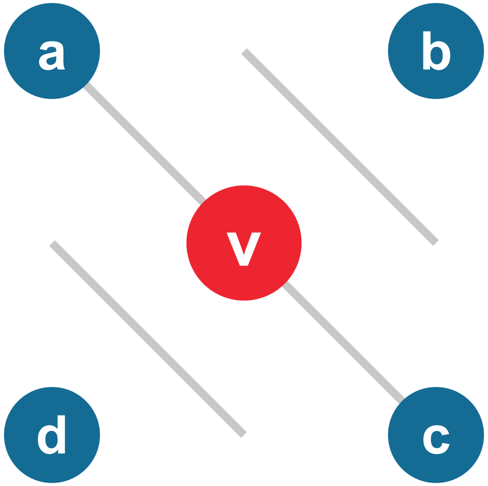
Voisinage

Définition

Le voisinage d'un sommet v est l'ensemble
des sommets directement connectés à v.

Notations

N(v) = voisinage ouvert (sans v)
N[v] = voisinage fermé (avec v)



N(v) = {a, b, c, d}

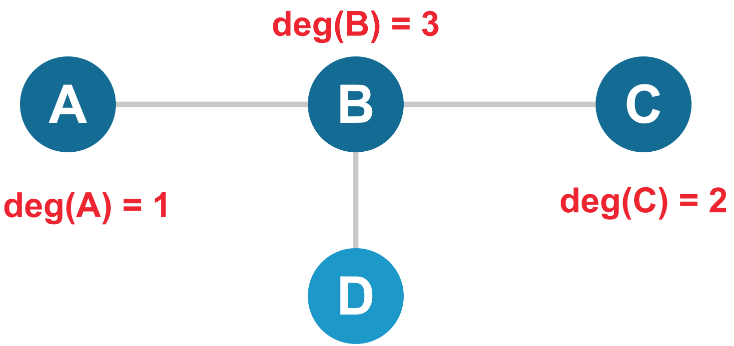
Degré d'un sommet

Définition

Le degré deg(v) d'un sommet v est le nombre d'arêtes incidentes à v.

Autrement dit : le nombre de voisins de v.

```
deg(v) = |N(v)|
```



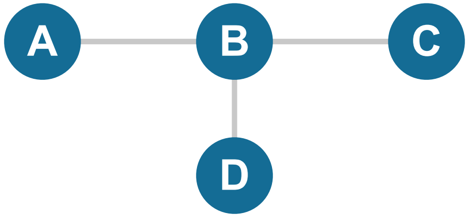
Propriété fondamentale

$$\Sigma \; deg(v) = 2|E|$$

📐 Interprétation

La somme des degrés de tous les sommets = 2 fois le nombre d'arêtes
Pourquoi ? Chaque arête est comptée 2 fois (une fois à chaque extrémité)

Exemple concret :



deg(A) + deg(B) + deg(C) + deg(D) = 1 + 3 + 1 + 1 = 6

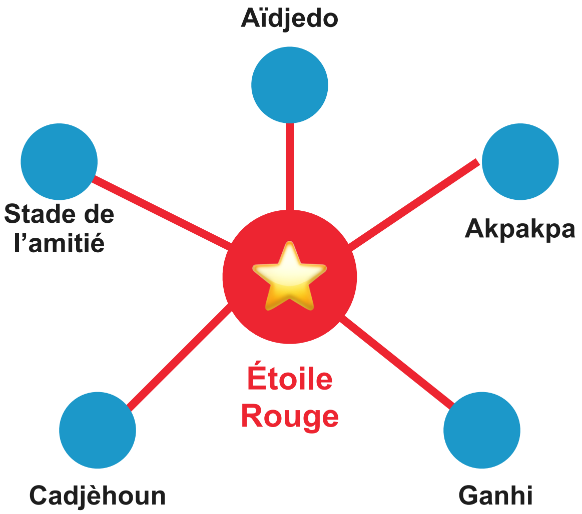
|E| = 3 arêtes → 2|E| = 6 ✓

Exemple : Carrefour Étoile Rouge

Un des plus grands carrefours de Cotonou

🚦 Description

5 routes qui y mènent :
- Route de Akpakpa
- Route de Ganhi
- Route de Cadjehoun
- Route du Stade de l'amitié
- Route de Aïdjedo



deg(Étoile Rouge) = 5

Chemin et Cycle

Chemin (Path)

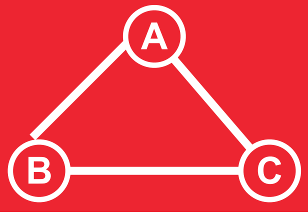
Séquence de sommets où chaque paire consécutive est reliée par une arête.

Exemple : A → B → C → D

Cycle

Chemin qui commence et finit au même sommet.

Exemple : A → B → C → A



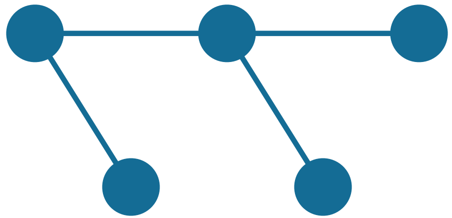
💡 **Important : Un cycle doit contenir au moins 3 sommets distincts**

Connexité

Définition

Un graphe est connexe s'il existe un chemin entre toute paire de sommets.

✓ **Graphe connexe**



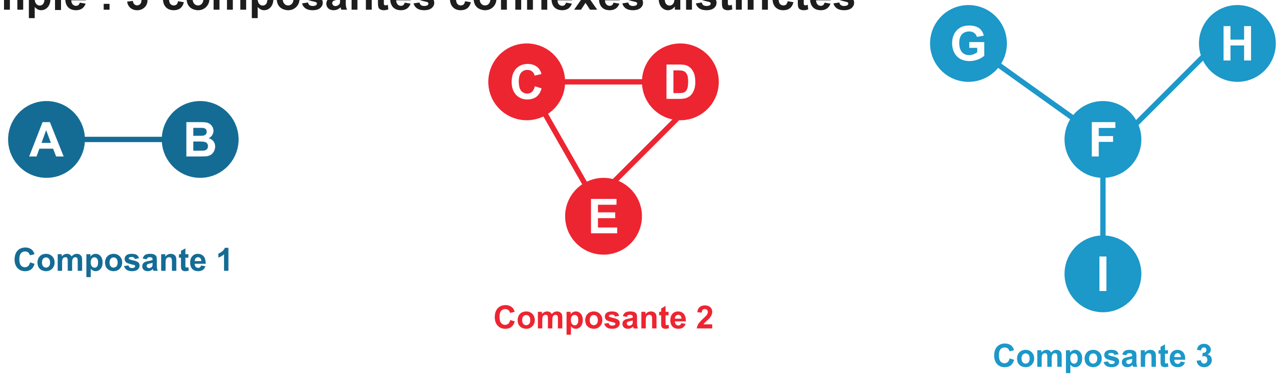
✗ **Graphe non connexe**

Composante connexe

Définition

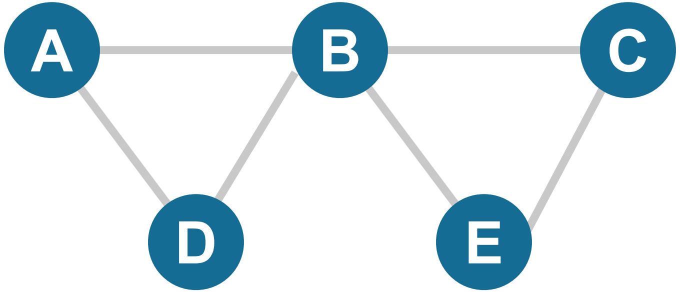
Sous-ensemble maximal de sommets tel qu'il existe un chemin entre chaque paire de sommets de cet ensemble.

Exemple : 3 composantes connexes distinctes



💡 Question minute

Calculez le degré de chaque sommet du graphe ci-dessous.
Vérifiez ensuite que $\Sigma deg(v) = 2|E|$



⏱ Temps : 30 secondes

Récapitulatif : Vocabulaire

Ce que nous avons appris

Graphe G=(V,E)	Ensemble de sommets V et d'arêtes E
Sommet	Nœud du graphe, entité du système
Arête	Connexion entre deux sommets
Voisinage N(v)	Ensemble des sommets directement connectés à v
Degré deg(v)	Nombre d'arêtes incidentes à v
Propriété	$\Sigma \deg(v) = 2\|E\|$
Chemin	Séquence de sommets reliés par des arêtes
Cycle	Chemin qui revient au point de départ
Connexité	Existence d'un chemin entre toute paire

PARTIE 4

Types de graphes

Orientés, pondérés, simples, complets...

Graphe orienté vs non-orienté

Non-orienté

- Arêtes bidirectionnelles
- (u,v) = (v,u)
- Relation symétrique

Orienté (Digraphe)

- Arcs directionnels
- u → v ≠ v → u
- Relation asymétrique

📱 Exemples

- Facebook : amitié
- WhatsApp : contact
- Routes bidirectionnelles

🐦 Exemples

- Twitter/X : follow
- Instagram : abonnement
- Routes à sens unique

Graphe pondéré

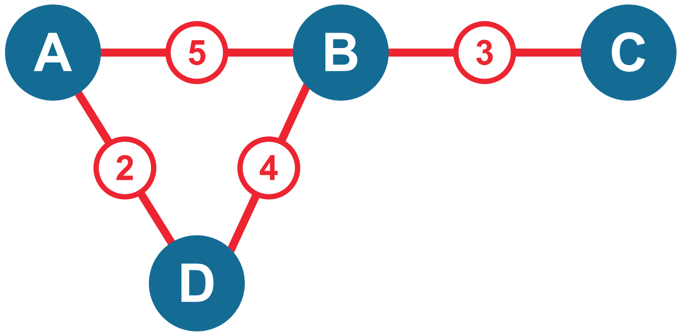
Définition

Graphe où chaque arête possède un poids (valeur numérique).

w: E → ℝ

📊 Applications

- Distance (GPS)
- Coût (réseau électrique)
- Temps de trajet
- Capacité d'un tuyau



Les nombres indiquent les poids des arêtes

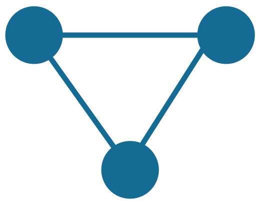
Graphe simple

Définition

Graphe sans :

- Boucles (arête reliant un sommet à lui-même)
- Arêtes multiples entre deux sommets

✓ Graphe simple



✓ Avec boucle

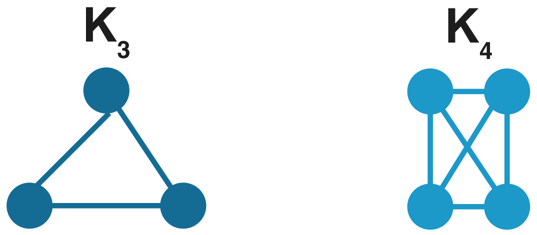
✓ Arêtes multiples

Graphe complet K_n

Définition

Graphe simple où chaque paire de sommets distincts est reliée par une arête.

$|E| = n(n-1)/2$



Nombre d'arêtes

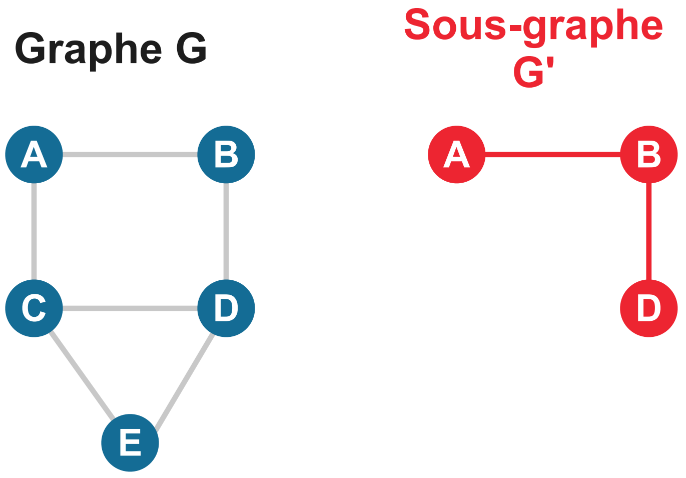
K_3 : 3 arêtes
K_4 : 6 arêtes
K_n : $n(n-1)/2$ arêtes

Sous-graphe

Définition

G' = (V', E') est un sous-graphe de G = (V, E) si :

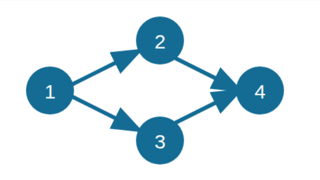
- $V' \subseteq V$
- $E' \subseteq E$



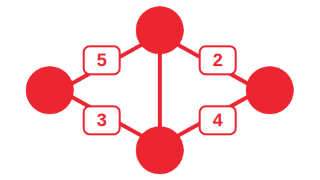
$V' = \{A, B, D\} \subseteq V = \{A, B, C, D, E\}$
$E' = \{(A,B), (B,D)\} \subseteq E$ (6 arêtes)

⏱️ Question Minute : Classifiez ces graphes !

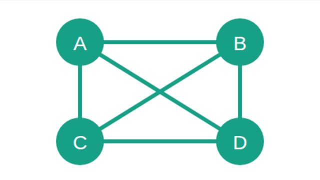
Graphe A



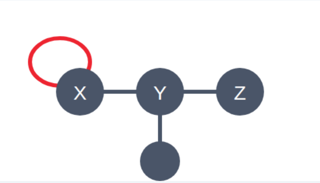
Graphe B



Graphe C



Graphe D



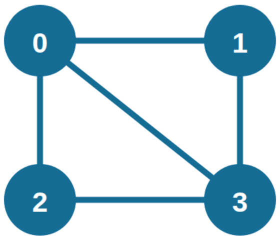
Classifiez :

- Orienté / Non-orienté
- Pondéré / Non-pondéré
- Simple / Non-simple
- Complet ?

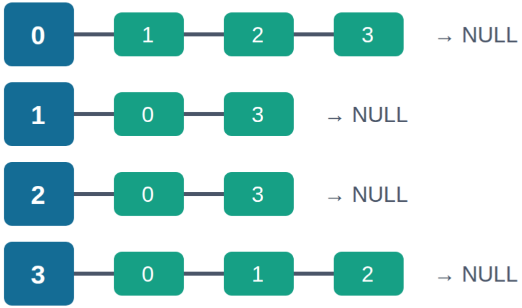
⏱️ 2 minutes !

📋 Liste d'adjacence : Structure

Graphe G



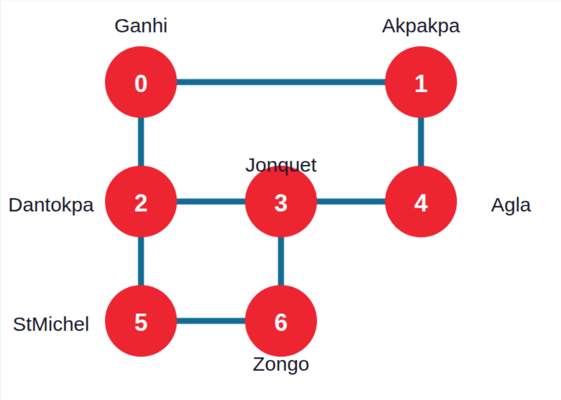
Liste d'adjacence



Principe : Tableau de listes chaînées. Chaque sommet pointe vers ses voisins.

🗺️ Liste d'adjacence : Exemple Cotonou

Réseau routier simplifié



7 sommets (quartiers) | **8 arêtes** (routes)

Liste d'adjacence en mémoire

adj[0] (Ganhi) → [1, 2]
adj[1] (Akpakpa) → [0, 4]
adj[2] (Dantokpa) → [0, 3, 5]
adj[3] (Jonquet) → [2, 4, 6]
adj[4] (Agla) → [1, 3]
adj[5] (StMichel) → [2, 6]
adj[6] (Zongo) → [3, 5]

📊 Espace mémoire

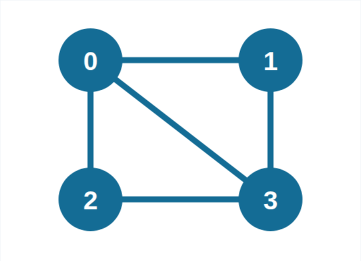
$O(V + E) = O(7 + 8) = \mathbf{O(15)}$

⚡ Avantage

Efficace pour les **graphes creux**

Matrice d'adjacence : Structure

Graphe G



Matrice d'adjacence

$$
\begin{array}{c|c}
 & \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\
\begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}
\end{array}
$$

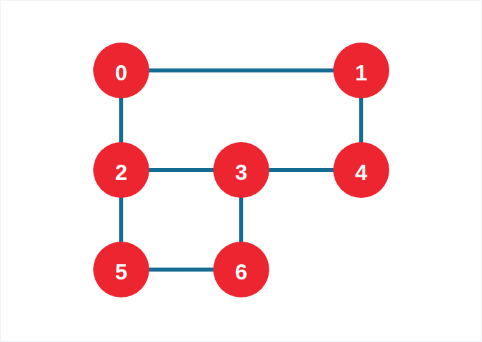
1 = Arête existe

0 = Pas d'arête

Principe : Tableau 2D où M[i][j] = 1 si arête (i,j) existe. **Symétrique** pour graphes non-orientés.

🗺️ Matrice d'adjacence : Exemple

Réseau routier (7 quartiers)



Matrice 7×7

$$
\begin{array}{c c}
 & \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\
\begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} &
\begin{bmatrix}
0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0
\end{bmatrix}
\end{array}
$$

📊 **Espace mémoire**

$O(V^2) = O(49)$ cases

⚡ **Avantage**

Vérifier arête en $O(1)$

⚠️ **Inconvénient**

Beaucoup de 0 (creux)

📐 Comparaison : Complexité en espace

📋 Liste d'adjacence

$O(V + E)$

Stocke uniquement les arêtes existantes

Exemple : V=1000, E=2000 → **3000** éléments

📊 Matrice d'adjacence

$O(V^2)$

Stocke toutes les paires possibles

Exemple : V=1000 → **1 000 000** cases

Graphe creux ($E \ll V^2$)

✅ **Liste gagne !**

Graphe dense ($E \approx V^2$)

≈ Équivalent

⏱️ Comparaison : Complexité en temps

Opération	📋 Liste	📊 Matrice
Ajouter une arête	O(1)	O(1)
Supprimer une arête	O(E)	O(1)
Tester si arête existe	O(V)	O(1) ✨
Lister voisins de v	O(deg(v)) ✨	O(V)
Parcourir tout graphe	O(V+E) ✨	$O(V^2)$

Liste : Idéale pour parcours (BFS/DFS) et lister voisins

Matrice : Idéale pour tester existence d'arêtes

🎯 Critères de choix : Quelle représentation ?

📋 Choisir LISTE si...

- Graphe **creux** (peu d'arêtes)
- Besoin de **parcourir les voisins**
- Algorithmes **BFS / DFS**
- Mémoire **limitée**
- Graphe **dynamique** (ajouts fréquents)

Exemples : Réseaux sociaux, Web, Routes

VS

📊 Choisir MATRICE si...

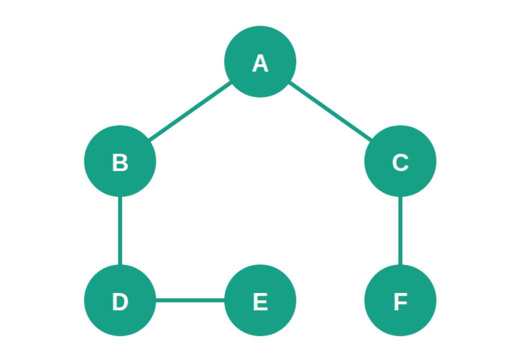
- Graphe **dense** (beaucoup d'arêtes)
- Besoin de **tester arêtes** souvent
- Opérations **matricielles**
- **Petit** nombre de sommets
- Graphe **complet** ou quasi-complet

Exemples : Jeux de plateau, Grilles, Cliques

💡 **Règle pratique :** Si $E < V^2/10$ → Liste. Si $E > V^2/2$ → Matrice. Entre les deux → dépend des opérations.

🕸 Exemple : Graphe CREUX (sparse)

Réseau social : 6 personnes



V = 6 sommets | **E = 5** arêtes

Comparaison des représentations

📋 **Liste d'adjacence**

A → [B, C]
B → [A, D]
C → [A, F]
D → [B, E]
E → [D]
F → [C]

10 éléments stockés

📊 **Matrice 6×6**

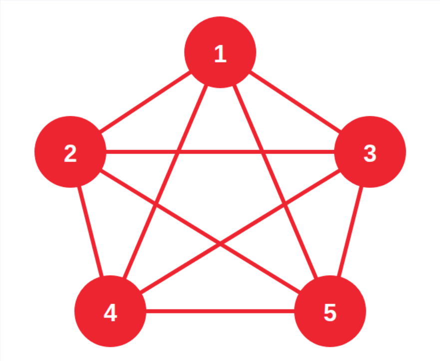
	A	B	C	D	E	F
A	0	1	1	0	0	0
B	1	0	0	1	0	0
C	1	0	0	0	0	1
D	0	1	0	0	1	0
E	0	0	0	1	0	0
F	0	0	1	0	0	0

36 cases (30 zéros !)

✅ **Verdict :** Liste d'adjacence gagne ! Ratio $E/V^2 = 5/36 \approx$ **14%** → Graphe creux

🔗 Exemple : Graphe DENSE (complet K_5)

Tournoi : 5 équipes (tous vs tous)



V = 5 | **E = 10** = n(n-1)/2

Comparaison des représentations

📋 **Liste d'adjacence**

1 → [2, 3, 4, 5]
2 → [1, 3, 4, 5]
3 → [1, 2, 4, 5]
4 → [1, 2, 3, 5]
5 → [1, 2, 3, 4]

20 éléments

📊 **Matrice 5×5**

	1	2	3	4	5
1	0	1	1	1	1
2	1	0	1	1	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	0

25 cases (20 uns !)

✅ **Verdict :** Matrice gagne ! Ratio E/V² = 10/25 = **40%** → Graphe dense + accès O(1)

PARTIE 6

Récapitulatif

de la séance

Ce que nous avons appris aujourd'hui

Synthèse de la séance 1

THÉORIE DES GRAPHES

$G = (V, E)$

TYPES

Orienté / Non-orienté

Pondéré / Simple / Complet

VOCABULAIRE

Sommet, Arête, Degré

Chemin, Connexité, Composante

REPRÉSENTATIONS

Liste d'adjacence : $O(V+E)$

Matrice d'adjacence : $O(V^2)$

APPLICATIONS

Réseaux sociaux, GPS

SBEE, Internet, Compilation

IMPLÉMENTATION

Structures en C/C++

TPE : Bibliothèque de graphes

PARTIE 7

TD1

Exercices dirigés

3 exercices 60 minutes Travail en binôme

Exercice 1.1 : Réseau routier de Cotonou

Contexte

Vous travaillez pour une application de navigation à Cotonou. Voici les distances entre 5 quartiers principaux :

Distances entre quartiers

Liaison	Distance
Ganhi ↔ Akpakpa	8 km
Ganhi ↔ Cadjèhoun	5 km
Akpakpa ↔ Fidjrossè	12 km
Cadjèhoun ↔ Fidjrossè	10 km
Cadjèhoun ↔ Godomey	7 km
Fidjrossè ↔ Godomey	6 km

Questions

1. Modéliser par un graphe (dessiner)
2. Orienté ou non-orienté ? Justifier
3. Pondéré ou non-pondéré ? Justifier
4. Donner V et E explicitement
5. Calculer |V| et |E|

Durée : 10 minutes

Exercice 1.2 : Organigramme d'entreprise

Contexte

Une entreprise béninoise a la structure hiérarchique suivante :

Structure hiérarchique

Directeur Général (DG) supervise :
→ Directeur Technique (DT)
→ Directeur Commercial (DC)

Directeur Technique supervise :
→ Ingénieur 1
→ Ingénieur 2

Directeur Commercial supervise :
→ Vendeur 1
→ Vendeur 2
→ Vendeur 3

Questions

1. Modéliser par un graphe (dessiner)
2. Orienté ou non-orienté ? Justifier
3. Pondéré ou non-pondéré ?
4. Combien de sommets et d'arêtes ?

Indice

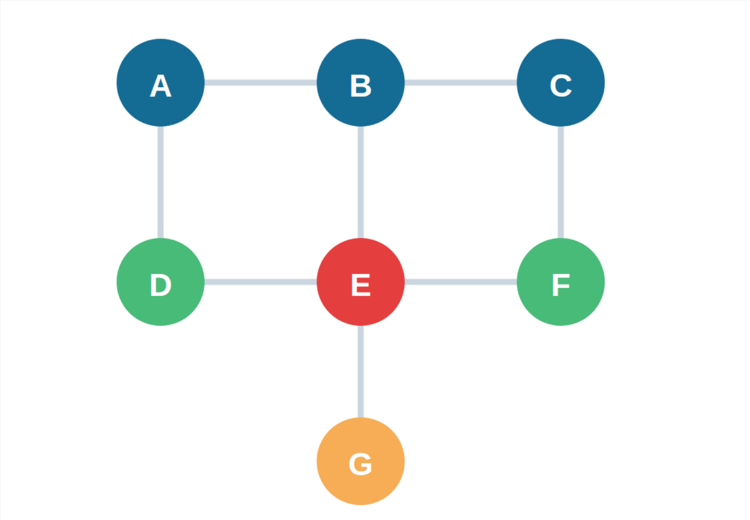
La relation "supervise" est-elle symétrique ?

Durée : 10 minutes

Exercice 2 : Vocabulaire fondamental

Le graphe G

Soit le graphe suivant :



Questions

1. Lister V et E
2. Calculer |V| et |E|
3. Calculer deg de chaque sommet
4. Vérifier : Σ deg(v) = 2|E|
5. Identifier N(E) et N[E]
6. Donner un chemin A → G
7. Donner un autre chemin A → G
8. Cycles de longueur 4 ?
9. Le graphe est-il connexe ?

Rappels

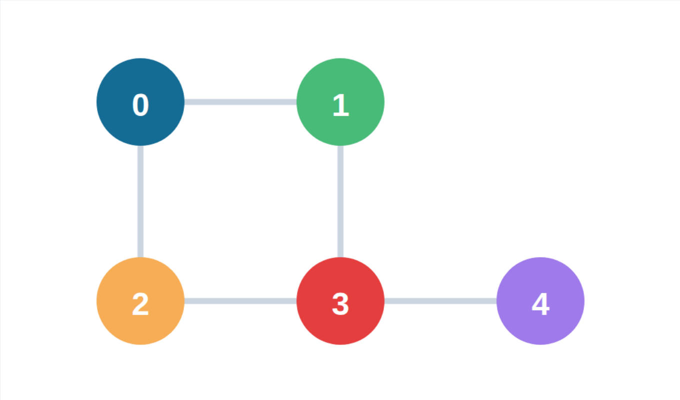
N(v) = voisinage ouvert (sans v)
N[v] = voisinage fermé (avec v)

Durée : 15 minutes

Exercice 3.1 & 3.2 : Liste et Matrice d'adjacence

Le graphe G

Soit le graphe suivant :



Question 3.1 : Liste d'adjacence

Écrire la liste d'adjacence de ce graphe

Format : 0 → [voisins de 0]

Question 3.2 : Matrice d'adjacence

Écrire la matrice d'adjacence (tableau 5×5)

0 si pas d'arête, 1 si arête

Rappel

Graphe non-orienté → matrice symétrique

Durée : 10 minutes (5 min chacun)

Exercice 3.3 : De la liste au graphe

Liste d'adjacence donnée

```
0 → [1, 3]
1 → [0, 2, 4]
2 → [1, 3]
3 → [0, 2, 4]
4 → [1, 3]
```

Méthode

1. Placer tous les sommets (0 à 4)
2. Tracer les arêtes selon la liste
3. Vérifier la cohérence

Questions

1. Dessiner le graphe correspondant
2. Le graphe est-il orienté ? Justifier
3. Calculer $|V|$ et $|E|$

Espace de travail

Utilisez votre feuille pour dessiner le graphe

Durée : 5 minutes

Exercice 3.4 : De la matrice au graphe

Matrice d'adjacence donnée

0	1		2		3			
0 [		0		1		1	0	]
1 [		0		0		1	1	]
2 [		0		0		0	1	]
3 [		0		0		0	0	]

Questions

1. Dessiner le graphe correspondant
2. Le graphe est-il orienté ?

   Comment le voit-on dans la matrice ?
3. Calculer |V| et |E|

Indice

Si mat[i][j] = 1, alors il y a un arc de i vers j

Espace de travail

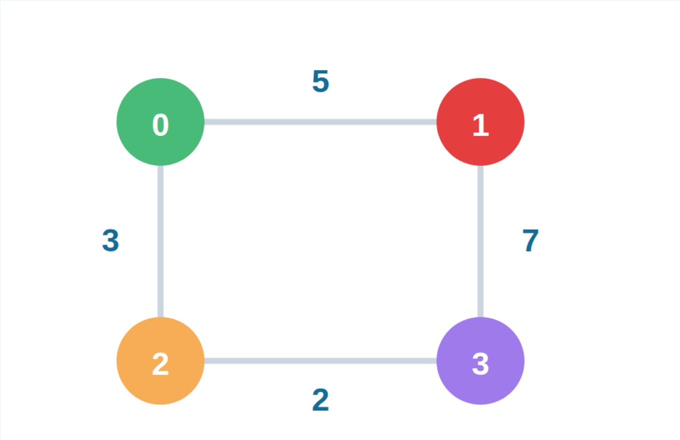
Dessinez le graphe avec des flèches pour les arcs

Durée : 5 minutes

Exercice 3.5 : Graphe pondéré

Le graphe pondéré G

Soit le graphe suivant :



Question 1 : Matrice d'adjacence

Écrire la matrice d'adjacence (4×4)

Utiliser ∞ pour l'absence d'arête

Question 2 : Liste d'adjacence

Écrire la liste d'adjacence avec poids

Format : 0 → [(voisin, poids), ...]

Exemple

`0 → [(1, 5), (2, 3)]`

Durée : 5 minutes

EXERCICE 4

BONUS

Réflexion

Pour aller plus loin | 10 minutes

Exercice 4.1 : Choix de représentation (BONUS)

Contexte algorithmique

Algorithme qui **parcourt tous les voisins de chaque sommet**.

1000
sommets

2000
arêtes

Question

Quelle représentation choisir ? Justifier en comparant :

1. Espace mémoire (formules O())
2. Complexité du parcours des voisins

Liste d'adjacence

Espace : $O(V + E)$

Parcours : $O(deg(v))$

Matrice d'adjacence

Espace : $O(V^2)$

Parcours : $O(V)$

Exercice 4.2 : Graphe complet (BONUS)

Question théorique

Un graphe **simple non-orienté** a **100 sommets**.

Quel est le nombre **maximum** d'arêtes possibles ?

Indice

Combien de **paires de sommets différents** ?

Graphe simple = pas d'arêtes multiples, pas de boucles

Formule à retrouver

Pour **n sommets**, le nombre max d'arêtes :

$$|E| = \mathbf{?}$$

Bravo !

Vous avez terminé le TD1

Compétences acquises

✓ Modéliser un problème réel en graphe

✓ Maîtriser le vocabulaire fondamental

✓ Convertir entre représentations

✓ Analyser les complexités

Prochaine étape : **TPE** (Implémentation en C++)