

IFT870/BIN710

Forage de données

Thème 6 : Forage de données de graphe

Aida Ouangraoua

Département d'informatique



Université de
Sherbrooke

Données sous forme de graphes

- Réseaux sociaux
- Structures moléculaires/biochimiques (protéines, ARN, complexes moléculaires)
- Réseaux d'expression génique
- Composés chimiques
- Graphes de flot de contrôle
- Graphes d'attaques en sécurité informatique
- Réseaux de communication

Forage de données de graphe

□ Recherche de motifs

- ❖ Fréquents
- ❖ Discriminatifs : score de Fisher, variance
- ❖ Recherche avec contrainte

□ Classification de graphes

- ❖ Basée sur des vecteurs d'attributs (**Exemple** : motifs fréquents)
- ❖ Basée sur des distances (**Exemple**: distance d'édition)

□ Partitionnement/Compression de graphes

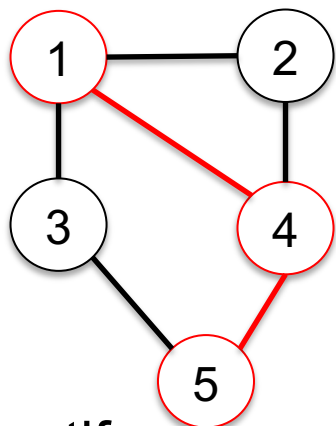
Recherche de motifs fréquents

- ❑ Séquences de transactions:

Ensemble de graphes $G_i = (V_i, E_i)$, $1 \leq i \leq n$ tel que V_i est l'ensemble des sommets de graphe G_i et E_i est l'ensemble des arêtes du graphe

- ❑ Un motif à k -arêtes de G_i est un sous-graphe connexe de G_i contenant k arêtes

- ❑ Exemple: si $V_i = \{1, 2, 3, 4, 5\}$ et $E_i = \{(1, 2), (1, 3), (1, 4), (2, 4), (3, 5), (4, 5)\}$, $G = (\{1, 4, 5\}, \{(1, 4), (4, 5)\})$ est un 2-arêtes de $G_i = (V_i, E_i)$



- ❑ Score d'un motif : pourcentage de graphes le contenant dans l'ensemble de graphes.

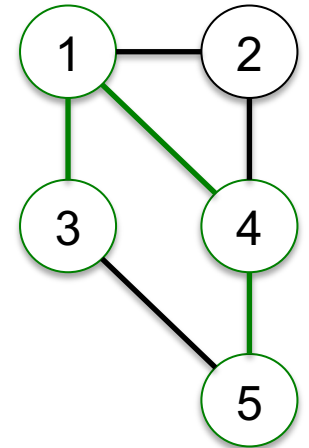
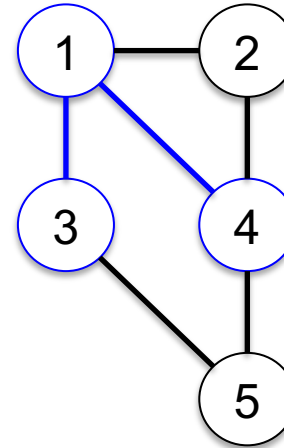
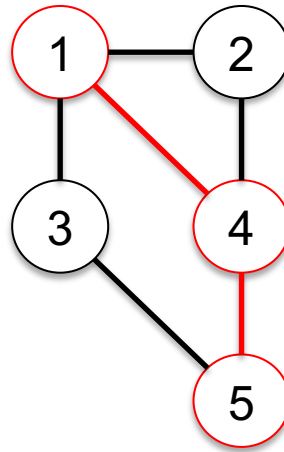
- ❑ Étant donné un seuil min_score , un motif est fréquent si son score est supérieur ou égal à min_score .

Recherche de motifs fréquents

- Algorithmes qui explorent l'espace de recherche de façon intelligente, en tenant compte de sa structure suivant différentes techniques
 - ❖ Génération des motifs candidats (k -arêtes \rightarrow $(k+1)$ -arêtes)
 - A-priori ou expansion de motifs
 - ❖ Ordre d'expansion/de recherche
 - En largeur ou en profondeur
 - ❖ Suppression des candidats dupliqués
 - ❖ Ordre de découverte
 - Ordre de génération
 - Chemin, arbre, graphe

Génération des motifs candidats (k-arêtes \rightarrow (k+1)-arêtes)

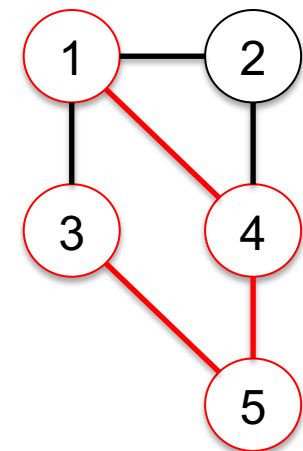
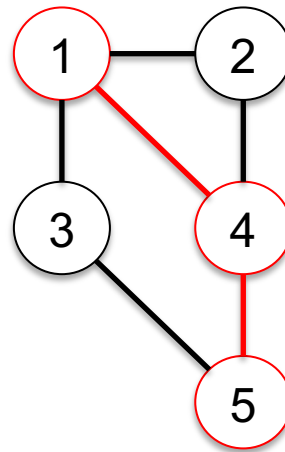
A-priori: si un motif est fréquent, alors tous ses sous-motifs sont fréquents.



Jointure de deux k-arêtes
partageant k-1 arêtes

un (k+1)-arêtes

Expansion: ajouter une arête à un k-arêtes pour obtenir un (k+1)-arêtes

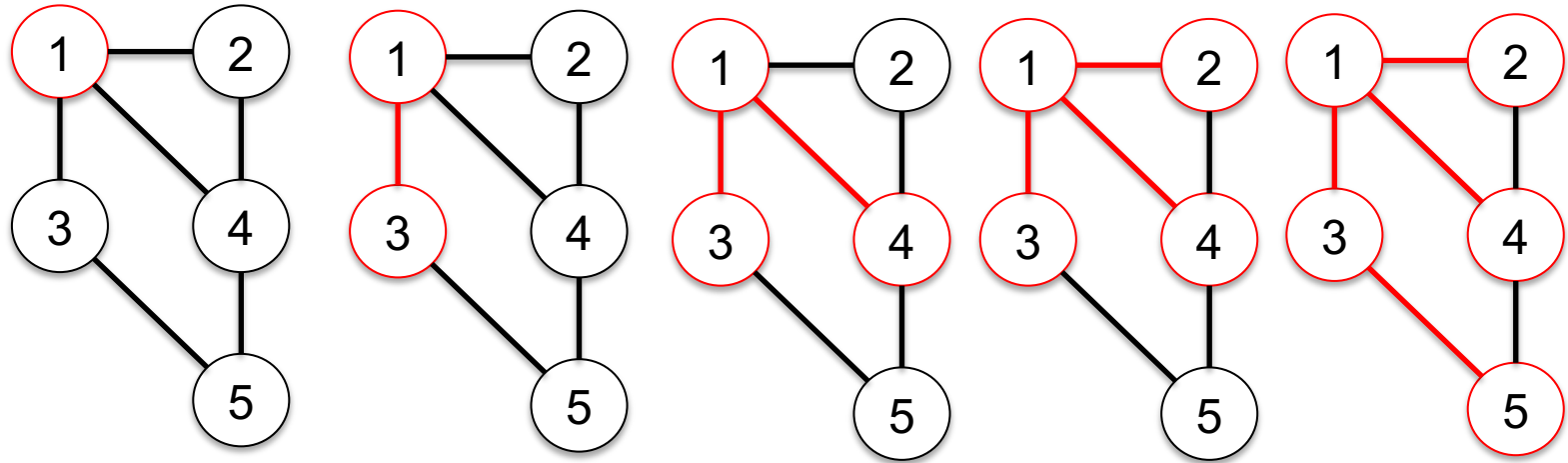


un k-arêtes
partageant k-1 arêtes

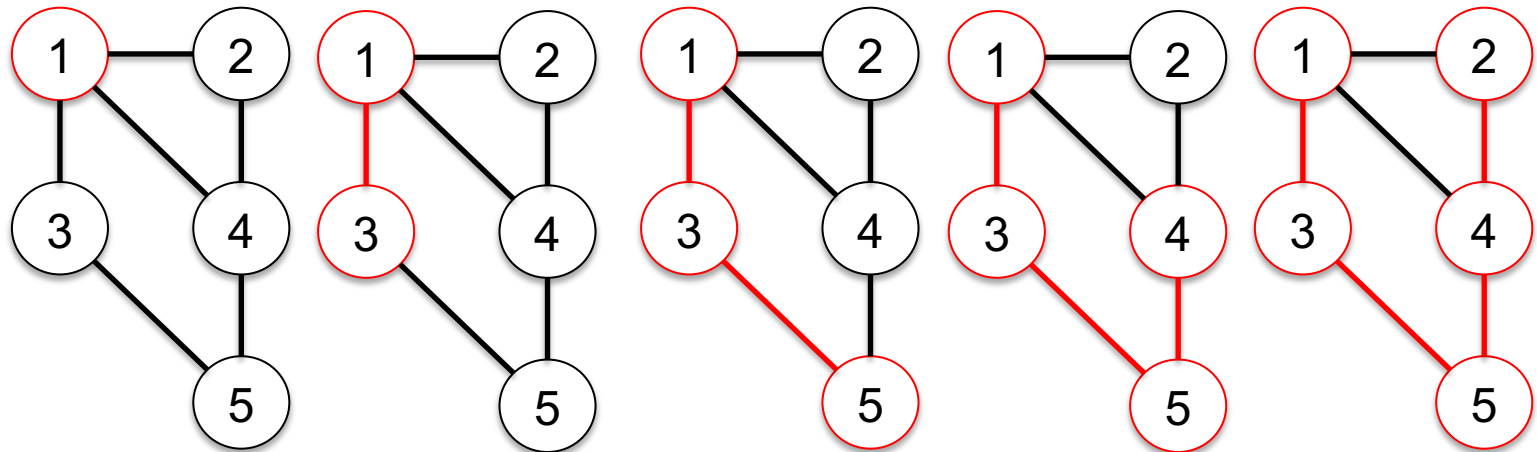
un (k+1)-arêtes

Ordre d'expansion

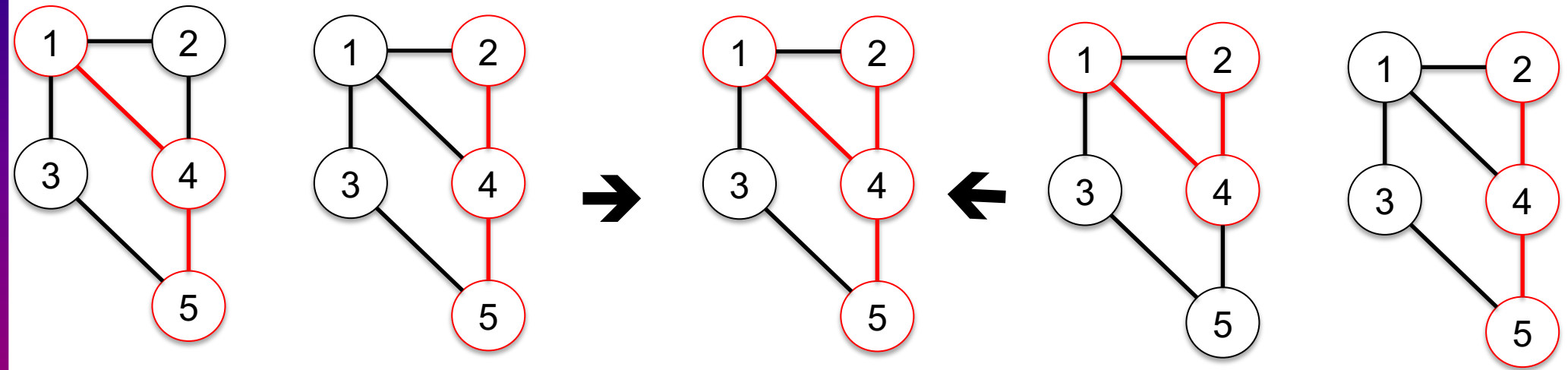
En largeur: ajouter tous les voisins d'un sommet avant d'ajouter les voisins de ces voisins



En profondeur: ajouter un voisin du dernier ajouté qui ait encore des voisins non visités

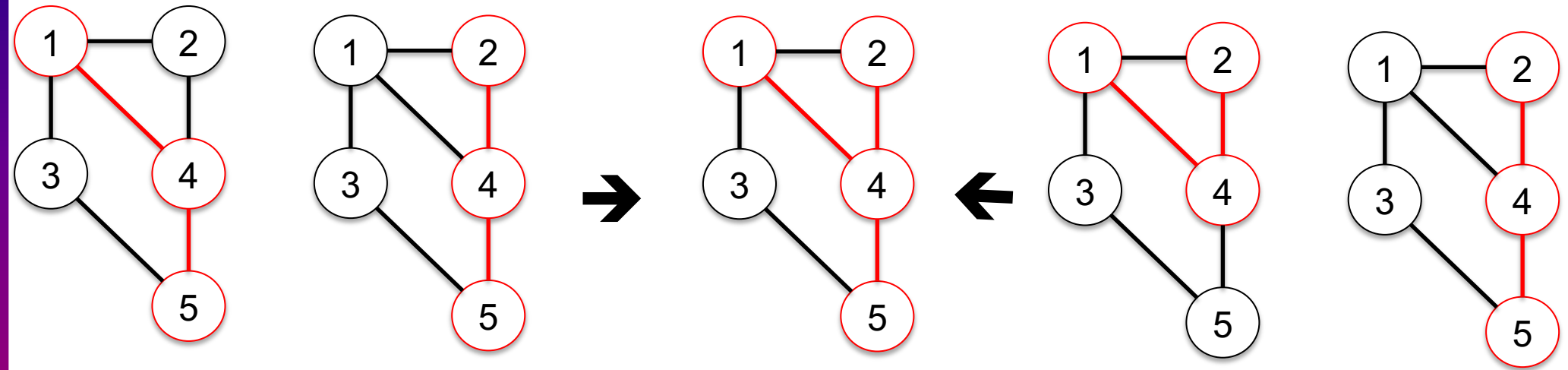


Suppression des candidats dupliqués



Un motif peut être généré plusieurs fois. Exemple: avec a-priori.
Il faut supprimer les dupliqués pour ne pas refaire les même calculs plusieurs fois

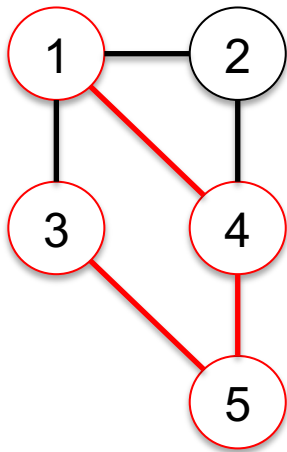
Suppression des candidats dupliqués



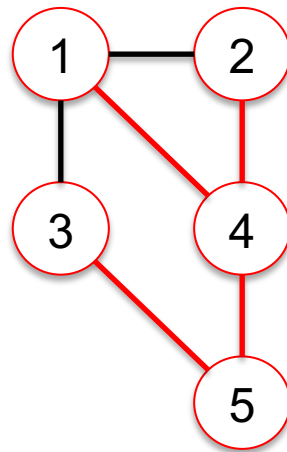
- Solution 1: comparer chaque nouveau motif à tous les motifs de même taille déjà généré (isomorphisme de graphe) → couteux en temps
- Solution 2: utiliser une fonction de hachage pour associer à chaque motif un identifiant, et comparer uniquement les identifiants
- Solution 3: définir un ordre sur l'ensemble des motifs de même taille et générer les motifs dans cet ordre.

Ordre de découverte

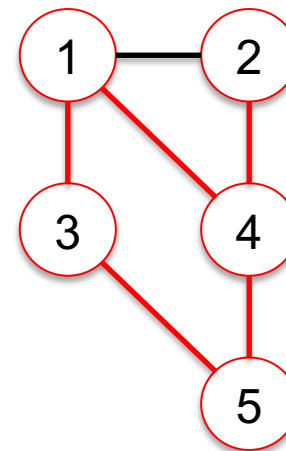
- ❑ Dans l'ordre de génération des motifs
- ❑ Suivant la structure des motifs: chemin → arbre → graphe



chemin:linéaire



arbre: pas de cycle



graphe

Gestion de la taille de l'espace de recherche

- ❑ Un graphe à n arêtes peut avoir 2^n motifs

- ❑ Problème:
 - ❖ Comment fixer le seuil de support ?
 - ❖ Quels sont les motifs conservés ?

- ❑ Solution:
 - ❖ Retenir un petit nombre de motifs représentatifs
 - Par clustering (plus pertinents)
 - Nécessite de définir une **mesure de distance entre motifs**
 - En gardant les plus fréquents (plus significatifs)

- ❑ Mesure de distance / similarité entre motifs
 - ❖ Intrinsèque, basée sur les motifs (exemple: distance d'édition)
 - ❖ Extrinsèque, basée sur les données (exemple: graphes partageant ce motifs)

Autres scores pour évaluer les motifs

- Score de Fisher : évalue si un motif de graphe apparaît plus souvent que prévu par hasard (test exact)
- G-test : compare les comptes observés de motifs à des comptes attendus sous l'hypothèse nulle pour évaluer si un motif est enrichi ou appauvri (test approximatif)
- Gain en information : mesure combien le motif apporte d'information pour distinguer un graphe réel d'un graphe aléatoire
- Cosine : mesure à quel point deux graphes ont des profils de motifs similaires

Recherche avec contrainte

- Degré
- Taille (#sommets, #arêtes)
- Densité (clique)
- Diamètre (plus long chemin entre deux sommets)
- Connectivité (#sommets, #arêtes)

Classification de graphes

- ❑ Basée sur des distances (Exemple: distance d'édition)
- ❑ Basée sur des vecteurs d'attributs (Exemple: motifs fréquents)

Classification basée sur des distances

- Distance d'édition entre deux graphes G_1 et G_2 :
Nombre minimum d'opérations élémentaires pour transformer un graphe en un autre.

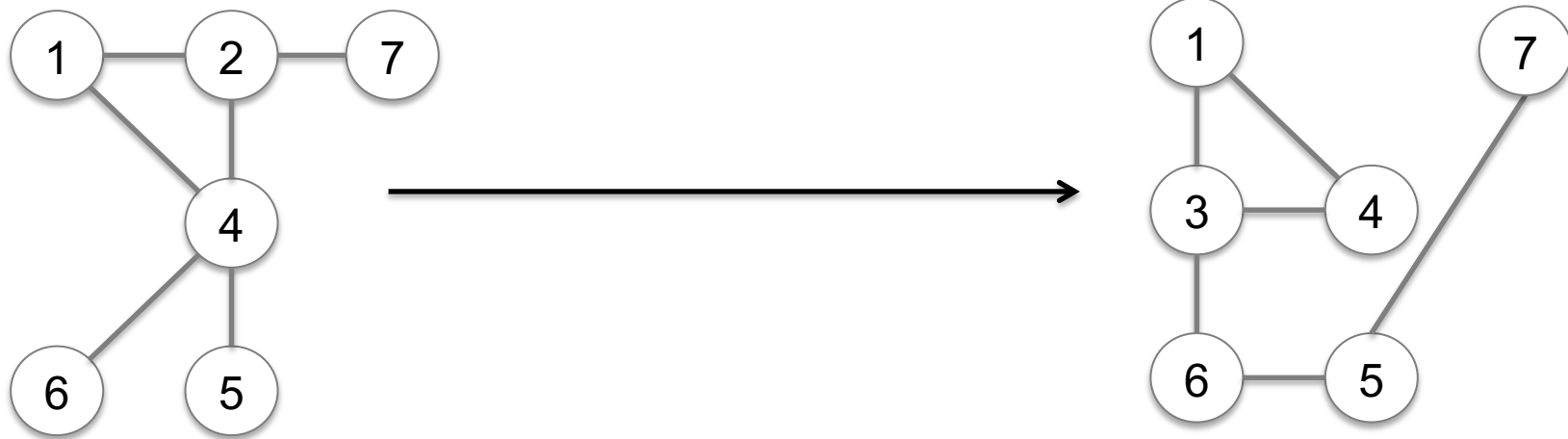
- Opérations élémentaires:
 - ❖ Insertion/Supression/Substitution de sommets
 - ❖ Insertion/Supression/Substitution d'arêtes
 - ❖ Division/Contraction d'arêtes

- Problème NP-complet, et difficile à approximer

- Solution :
 - ❖ Calcul du plus court chemin dans une collection de graphes dont les sommets sont des graphes incluant G_1 et G_2 , et il existe une arête entre deux graphes G et G' si une seule opération élémentaire sépare G et G'
 - ❖ Algorithme de recherche A^*

Classification basée sur des distances

- ❑ Distance d'édition simple (coûts unitaires)
- ❑ Distance d'édition pondérée (un coût par type d'opération)
- ❑ Distance d'édition généralisée (coût dépendant des étiquettes)



Classification basée sur des vecteurs d'attributs

- ❑ **Objectif:** une représentation vectorielle des graphes pour utiliser des méthodes de classifications usuelles comme les arbres de décision

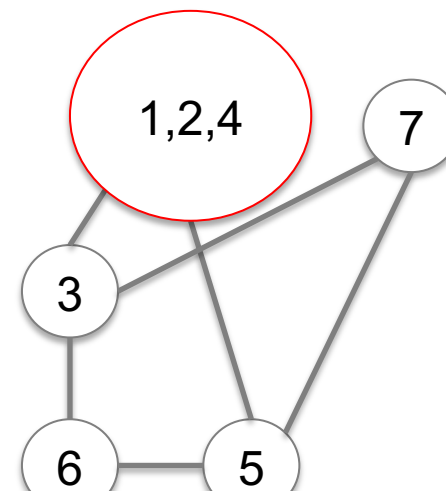
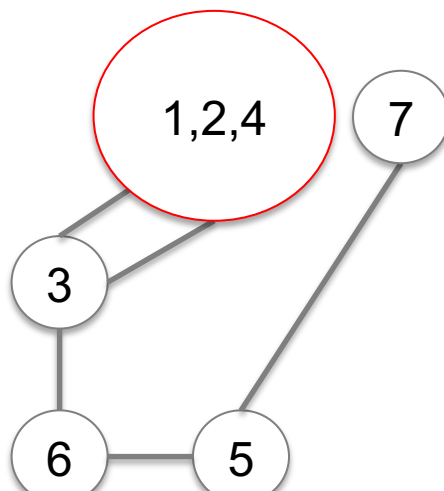
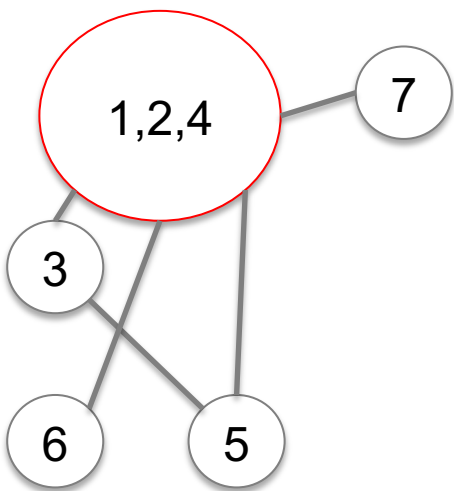
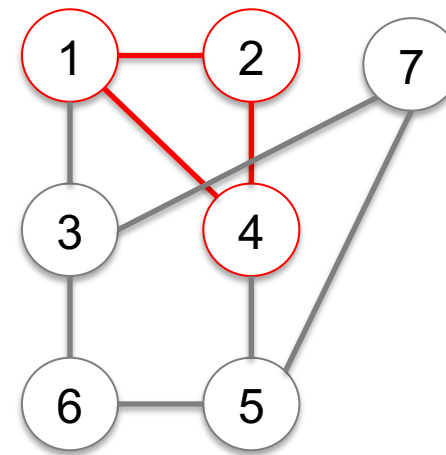
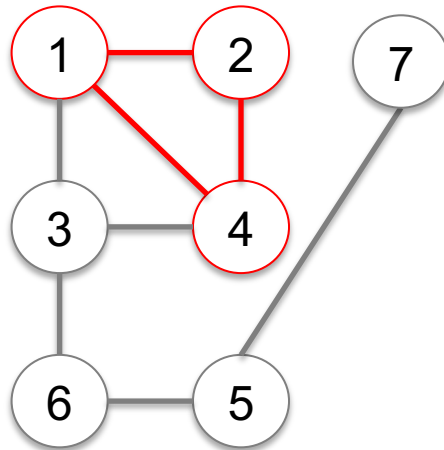
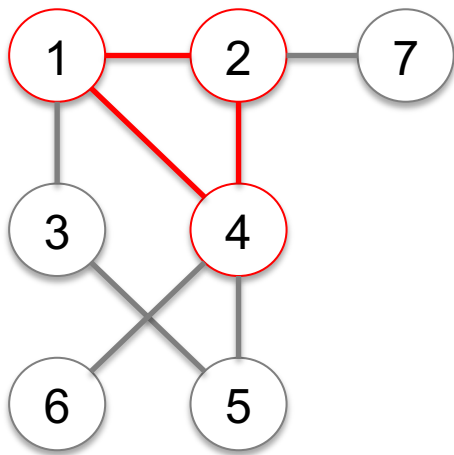
- ❑ Vecteurs des fréquences de motifs utilisés pour représenter les graphes
 - ❖ Motifs fréquents à k-arêtes

 - ❖ Sous-structures locales. **Exemple:** chemin à k arêtes, cliques.

 - ❖ Motifs issus du domaine de connaissance

Partitionnement / Compression de graphes

- ❑ Trouver des motifs communs (ou très similaires)
- ❑ Remplacer chaque motif commun par un sommet



Cas particulier des arbres

- Problèmes plus faciles que ceux des graphes, généralisation des solutions pour les séquences
 - ❖ Recherche de sous-arbres fréquents
 - ❖ Recherche de sous-arbres communs
 - ❖ Classification d'arbres
 - Distance d'édition d'arbres (non-orientés, ordonnés, non-ordonnés semi-ordonnés) → complexité $O(n^3)$
 - Modèles de covariance (extension des HMMs pour des arbres d'ARN)
 - ❖ Compression d'arbres

Références

- [1] Diane J. COOK, Lawrence B. HOLDER (Eds): *Mining Graph Data*, , John Wiley & Sons, 2006.
- [2] Jiawei HAN, Micheline KAMBER, Jian PEI. *DataMining: Concepts and Techniques (Third edition)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.
- [3] Adreas C. MÜLLER et Sarah GUIDO : *Introduction to Machine Learning with Python*. O'Reilly Media, Inc., Sebastopol, CA, 2017.