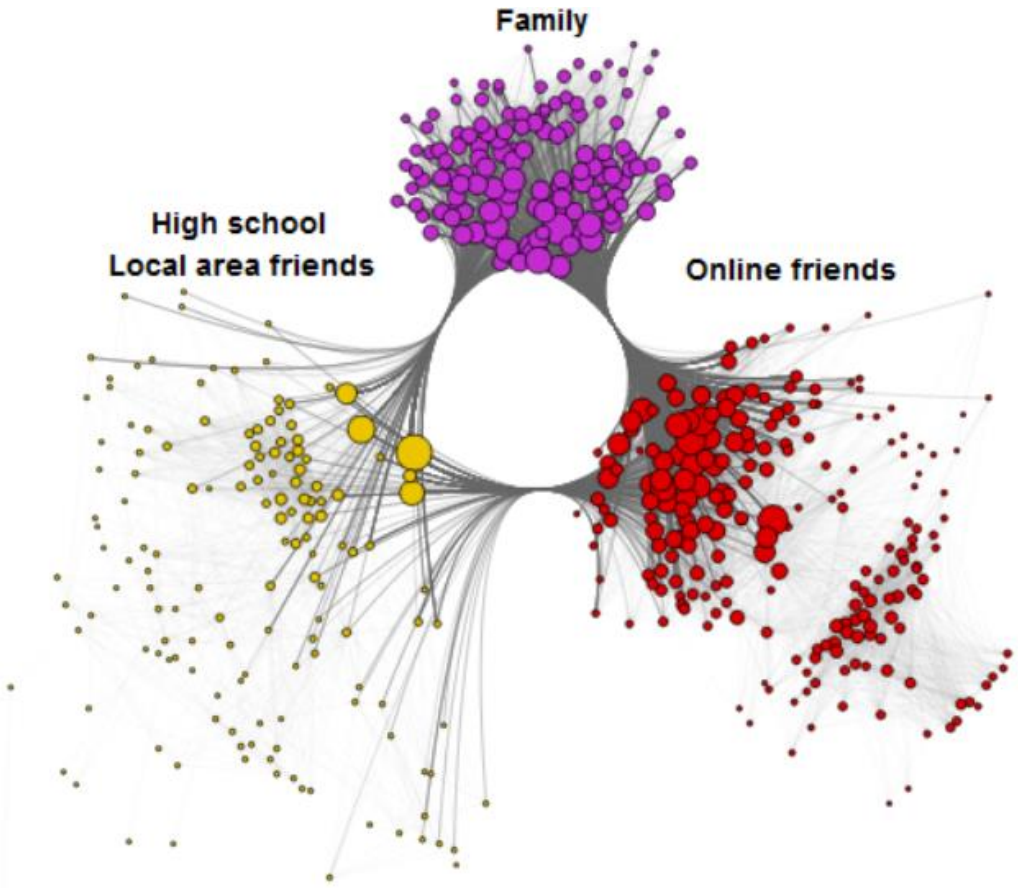
Part IV

Comunity Detection

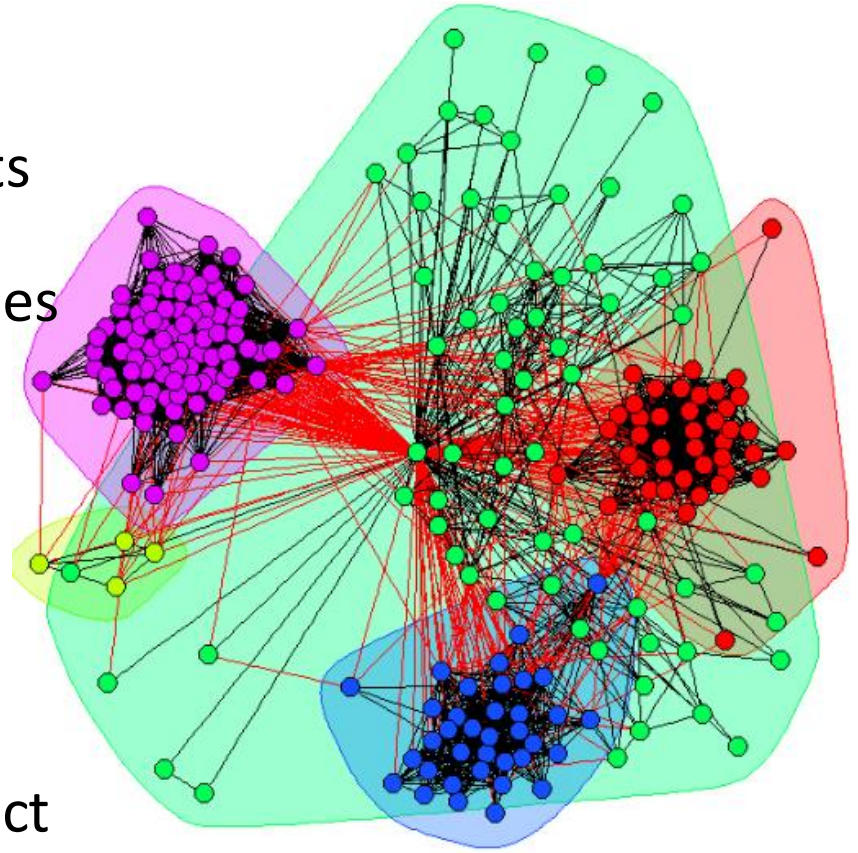
Comunity detection – definitions

- Les communautés sont une propriété de nombreux réseaux.
  - Exemple interactions Facebook
- Communauté: sous-ensemble de nœuds qui sont **étroitement** liés, c'est-à-dire qu'il existe entre eux un nombre significatif de connexions, plus important que les liens qu'ils entretiennent avec les autres membres du réseau.



Comunity detection – properties

- La détection des communautés est l'une des tâches les plus **importantes** en SNA consistant à trouver les divisions naturelles d'un réseau en groupes de sommets
- Comunity detection vs **clustering** (Machine Learning): , partage à la base de liens ou à la base de caractéristiques individuelles.
- Dans un réseau à **grande échelle**(millions de nœuds et d'arêtes), détecter les communautés dans de tels réseaux devient une tâche fastidieuse (NP-complet).
- Les communautés peuvent être bien séparées (techniques classiques) mais le plus souvent sont **recouvrantes** (techniques avancées).
- Le changement continu de la structure de réseau (aspect **dynamique**) complique davantage la tâche de détection.



Comunity detection – why ?

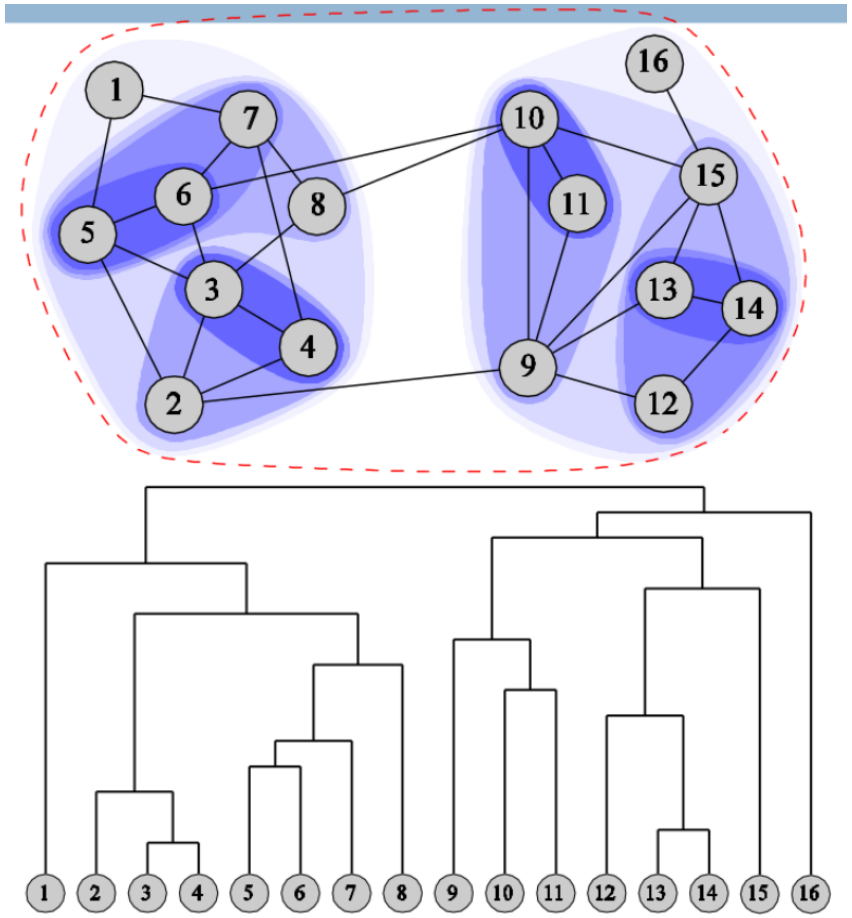
- Lors de l'analyse de différents réseaux, il peut être important de découvrir des communautés à l'intérieur de ceux-ci.
- La détection de communauté peut être utilisée en ML pour détecter des groupes ayant des propriétés similaires pour diverses raisons .
- découvrir des personnes ayant des intérêts communs et les maintenir étroitement connectées.
- On peut également se servir des communautés pour élaborer des systèmes de recommandations.
- Les protéines impliquées dans la même maladie ont tendance à interagir les unes avec les autres
- Etudier les parties politiques à travers leurs communautés

Comunity detection – techniques

- Les techniques et les algorithmes de DC sont divers et leur classification peut se faire selon plusieurs visions
- Méthodes classiques: graph partionning, spectral clustering, ...
- Méthodes hiérarchiques: Agglomerative and divisive algorithms
- Méthodes d'optimisation: Greedy algorithms, Louvain algorithm, …
- Méthodes de chevauchement: Clique percolation, Label propagation algorithm, …
- Méthodes dynamiques: Potts model, Random walk technique, …
- …

Ascending Hierarchical Clustering (Agglomerative Nesting-AGNES)

- Principe
  - il construit une hiérarchie de clusters à partir des nœuds d'un réseau donné.
  - Initialement, chaque nœud est affecté à son propre cluster.
  - Ensuite, les deux clusters **les plus proches** sont fusionnés dans le même cluster.
  - Ce processus est répété jusqu'à ce qu'il ne reste qu'un seul cluster.



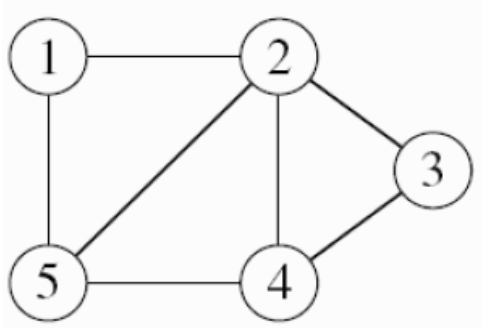
Ascending Hierarchical Clustering (Agglomerative Nesting-AGNES)

- Pour déterminer les nœuds/clusters les plus proches une mesure de distance est nécessaire
  - Distance euclidienne au carré: $d_{ij} = k_i + k_j - 2n_{ij}$
  - la mesure de similarité cosinus: $\sigma_{ij} = \frac{n_{ij}}{\sqrt{k_i}\sqrt{k_j}} = \frac{n_{ij}}{\sqrt{k_i k_j}}$
  - le coefficient de corrélation de Pearson standard: $r_{ij} = \frac{n_{ij} - \frac{k_i k_j}{N}}{\sqrt{k_i - \frac{k_i^2}{N}}\sqrt{k_j - \frac{k_j^2}{N}}}$

    Où, **nij**: nbr de voisins en commun, **ki**: degré de i, **N**: nbr de nœuds
- Etendre la distance pour le cas de clusters
  - clustering à liaison unique (min)
  - clustering à liaison complète (max)
  - clustering à liaison moyenne (moyenne)

Ascending Hierarchical Clustering (Agglomerative Nesting-AGNES)

- Algorithme
  1. Chaque sommet est dans une communauté
  2. Calculer la distance entre chaque paire de communautés (matrice de distances)
  3. Fusionner les deux plus proches
  4. Arrêter si tout est fusionné sinon revenir à (2)
- Exemple
  - Nous voulons effectuer un clustering hiérarchique à liaison unique en utilisant la distance euclidienne au carré pour mesurer la distance entre les nœuds



Ascending Hierarchical Clustering (Agglomerative Nesting-AGNES)

- Exemple
  - Les degrés des cinq nœuds sont: k1 = 2, k2 = 4, k3 = 2, k4 = 3, k5 = 3
  - La matrice des voisins en communs (nij):
  - En applicant la formule de distance, on obtiendra la matrice de distance (dij):

$$[n_{ij}] = \begin{bmatrix} 2 & 1 & 1 & 2 & 1 \\ 1 & 4 & 1 & 2 & 2 \\ 1 & 1 & 2 & 1 & 2 \\ 2 & 2 & 1 & 3 & 1 \\ 1 & 2 & 2 & 1 & 3 \end{bmatrix} \quad D = [d_{ij}] = \begin{bmatrix} 0 & 4 & 2 & 1 & 3 \\ 4 & 0 & 4 & 3 & 3 \\ 2 & 4 & 0 & 3 & 1 \\ 1 & 3 & 3 & 0 & 4 \\ 3 & 3 & 1 & 4 & 0 \end{bmatrix}.$$

clusters	1	2	3	4	5
1	0	4	2	**1**	3
2	4	0	4	3	3
3	2	4	0	3	**1**
4	1	3	3	0	4
5	3	3	1	4	0

(1)

clusters	{1,4}	{3,5}	2
{1,4}	0	**2**	3
{3,5}	2	0	3
2	3	3	0

(2)

clusters	{{1,4},{3,5}}	2
{{1,4}{3,5}}	0	**3**
2	3	0

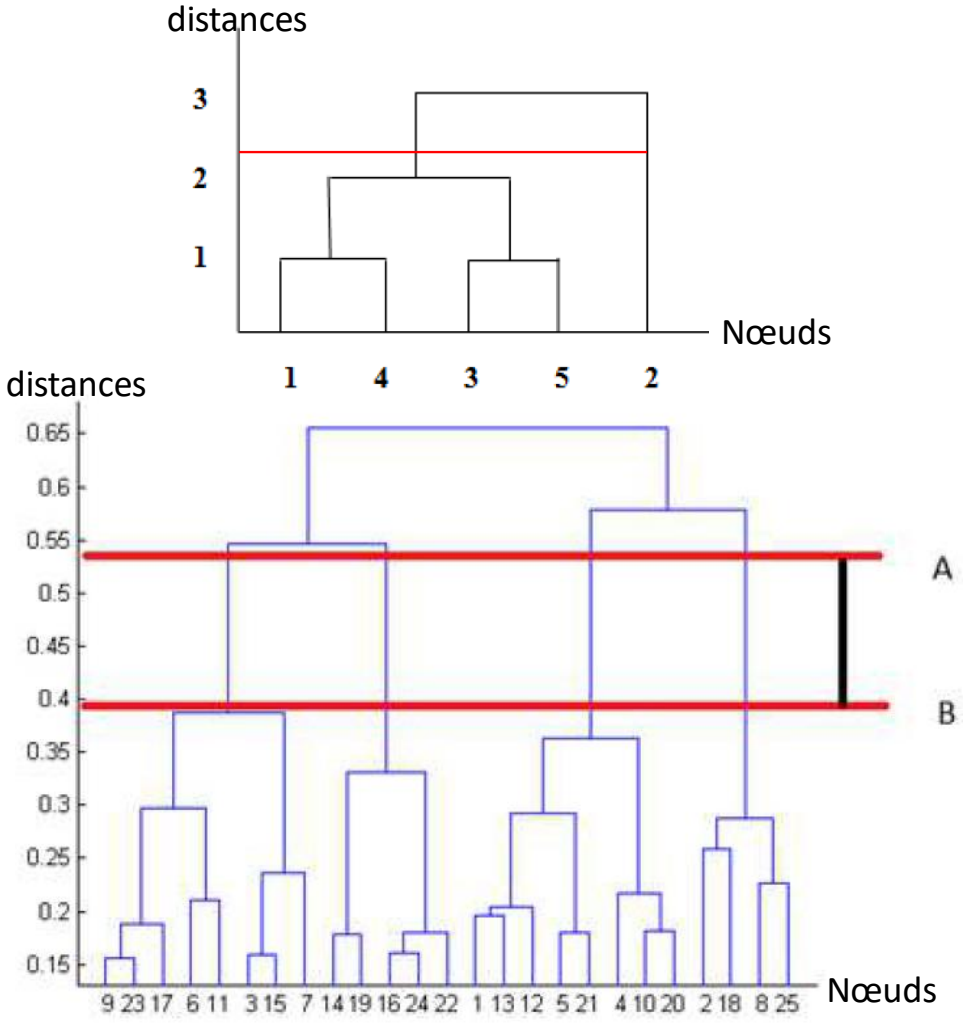
(3)

clusters	{{{1,4},{3,5}},2}
{{{1,4},{3,5}},2}	**0**

(4)

Ascending Hierarchical Clustering (Agglomerative Nesting-AGNES)

- Dendrogramme



Louvain method

- principe
  - La méthode de détection de communauté Louvain est une méthode d'optimisation
  - Son principe est **l'optimisation de la modularité** au fur et à mesure que l'algorithme progresse. L'optimisation de cette valeur conduit théoriquement au meilleur regroupement possible.
  - La modularité est une valeur d'échelle (de -0,5 à 1) mesurant la densité relative des arêtes à l'intérieur des communautés par rapport aux arêtes extérieures. Elle mesure la qualité des communautés détectées.
  - Louvain commence par trouver les petites communautés en optimisant la modularité sur tous les nœuds, puis chaque petite communauté est regroupée en un nœud et l’étape précédente est répétée.

Louvain method

- La modularité est donnée par:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

m: nbr d'arrêtes/somme des poids de toutes les arrêtes
Aij: poids de l'arrête (i,j)
ki: degré de i/somme des poids des arrêtes de i
ci: communauté de i
δ(ci,cj) = 1 si x=y, δ(ci,cj) = 0 sinon

Louvain method

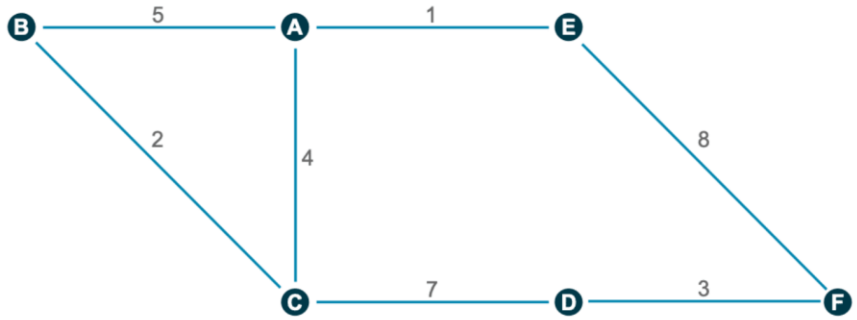
- Algorithme
  1. Chaque nœud du réseau est affecté à sa propre communauté.
  2. Calculer la modularité pour chaque **changement** de nœud i vers la communauté de chacun de ses **voisins** j:

$$\delta Q = \left[A_{ij} - \frac{k_i k_j}{m} \right]$$

  3. Une fois Q est calculée pour tous les voisins de i, i est placé dans la communauté qui a entraîné la plus grande augmentation de Q. Ce processus est appliqué à tous les nœuds jusqu'à ce qu'aucune augmentation de la modularité ne puisse se produire.
  4. regrouper tous les nœuds d'une même communauté et construit un nouveau réseau où les nœuds sont les communautés de la phase précédente. Les externes deviennent des liens valués.
  5. Terminer si Q atteint son max, sinon revenir à (1)

Louvain method

- Exemple: soit le réseau suivant



A → **B**: $Q_{AB} = 5 - \frac{10*7}{30} = 2.667$

B → C: $Q_{BC} = 2 - \frac{7*13}{30} = -1.033$

A → C: $Q_{AC} = 4 - \frac{10*13}{30} = -0.333$

C → **D**: $Q_{CD} = 7 - \frac{13*10}{30} = 2.667$

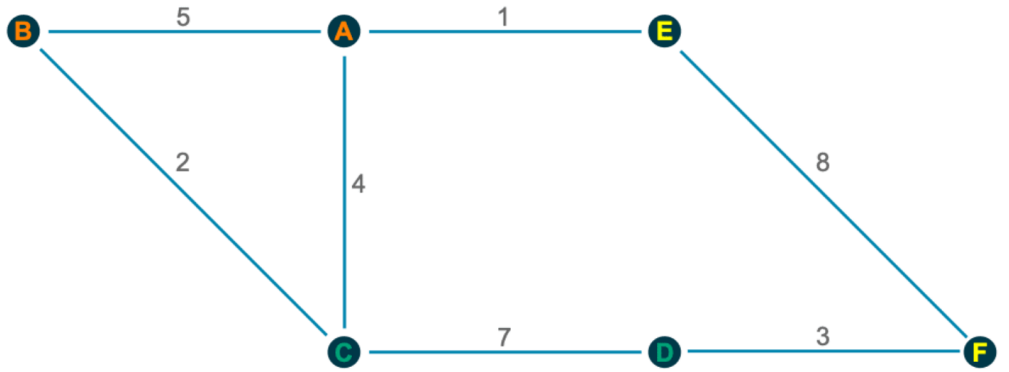
E → **F**: $Q_{AB} = 8 - \frac{9*11}{30} = 4.7$

A → E: $Q_{AE} = 1 - \frac{10*9}{30} = -2$

D → F: $Q_{CD} = 3 - \frac{10*11}{30} = -0.667$

Louvain method

- Exemple



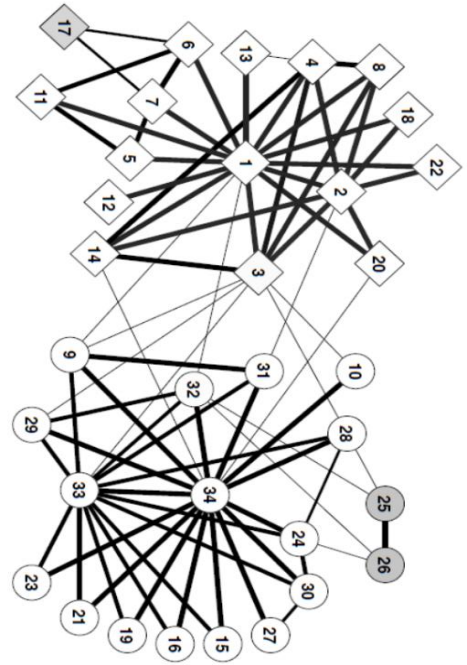
Orange → Green: $Q_{Or,Gr} = 6 - \frac{7*9}{10} = -0.3$

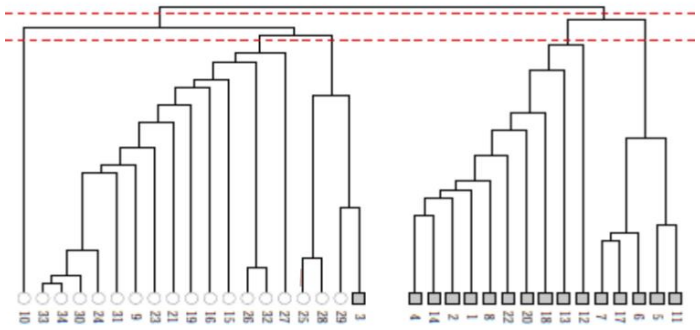
Orange → Yellow: $Q_{Or,Ye} = 1 - \frac{7*4}{10} = -1.8$

Green → Yellow: $Q_{Gr,Ye} = 3 - \frac{9*4}{10} = -0.6$

Clustering by division (Girvan & Newman)

- principe
  - Il repose sur l'idée qu'un **lien** qui se trouve fréquemment sur les plus courts chemins entre les nœuds du graphe, ne se trouve pas au sein d'une communauté donnée, mais plutôt qu'il relie des communautés distinctes.
  - Le lien avec la plus grande valeur de centralité d'intermédiarité est supprimé en premier
  - En retirant progressivement le lien qui a la plus forte centralité, on obtient un découpage en blocs de notre réseau.



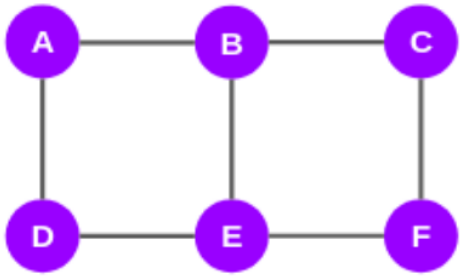


Clustering by division (Girvan & Newman)

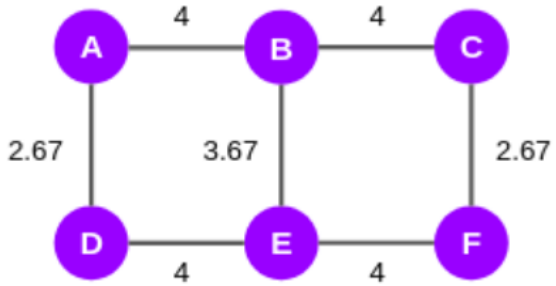
- L'algorithme de Girvan-Newman peut être divisé en quatre étapes principales:
  1. Pour chaque lien du réseau, calculez la centralité d'intermédiarité des liens.
  2. Supprimez le lien avec la centralité d'intermédiarité la plus élevée.
  3. Calculez la centralité d'intermédiarité pour chaque lien restant.
  4. Répétez les étapes 2 à 4 jusqu'à ce qu'il ne reste plus de liens.
- Le découpage itératif dans l'algorithme est éventuellement dépondant de la qualité du partitionnement fait. Cette qualité est calculée par la **Modularité** (algo Louvain)

Clustering by division (Girvan & Newman)

- Exemple: soit le réseau simple suivant

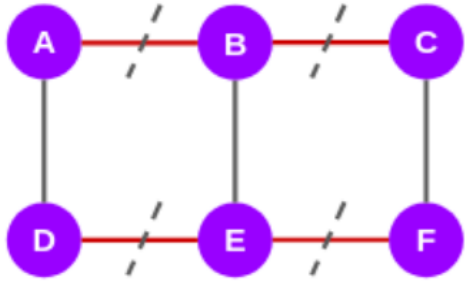


(1) On commence par le calcul de la centralité d'intermédiarité pour toutes les arêtes

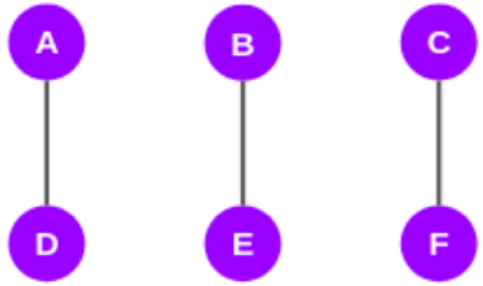


La modularité Q =
Q(A,B) + Q(A,C) + Q(A,D) + Q(A,E) + Q(A,F) + Q(B,C) + Q(B,D) + Q(B,E) + Q(B,F) + Q(C,D) + Q(C,E) + Q(C,F) + Q(D,E) + Q(D,F) + Q(E,F)

(2) Supprimer les arêtes ayant la valeur max=4 de centralité d'intermédiarité



(3) Trois communautés résultantes, recalculer la centralité …



La modularité Q =
Q(A,D) + Q(B,E) + Q(C,F),
sa prochaine valeur est nulle