Corrections des TD 1 à 6

Remarques importantes

- Inutile d'imprimer ces corrigés : ce ne sont pas des documents autorisés lors des examens.
- On ne comprend vraiment qu'en faisant soi-même. Lire la correction d'un exercice sans avoir fait l'exercice est une perte de temps ; c'est même pire, puisque potentiellement on perd une occasion d'apprendre quelque chose.

Correction TD 1 : Bases

Remarque : quand on demande quel est le type *d'un graphe, la réponse est de la forme : graphe orienté ou non, avec ou sans arêtes multiples, avec ou sans boucles, valué ou non...*

Exercice 1 (Graphes sur le web)**.** On considère les graphes suivants :

— le graphe des *followers* sur Instagram (ou Twitter, Bluesky, Threads...) (chaque compte peut suivre n'importe quel autre compte)

— le graphe des ami·es sur Facebook (deux personnes ne peuvent être amies que si elles sont d'accord toutes les deux)

— le graphe de liaison des pages Wikipédia — une page A étant *liée* à une page B si A contient (au moins) un lien vers B.

— le graphe représentant tous les liens hypertextes (éléments `a`) du web. Le graphe doit faire apparaître, pour chaque lien, l'URL où on peut le trouver, l'URL vers laquelle il mène, et son texte (ce sur quoi on peut cliquer).

Répondre aux questions suivantes pour chacun des graphes.

1. que représentent les sommets et les arêtes/arcs ?
2. de que type de graphe s'agit-il ?
3. à quoi correspond l'ordre du graphe ?
4. à quoi correspond le degré d'un sommet (ou les degrés entrant et sortant) ?

Instagram. les sommets sont des comptes Instagram, un arc relie un compte A à un compte B ssi A suit B. Graphe orienté, sans arêtes multiples, sans boucles. ordre = nombre de comptes Instagram. degré entrant d'un sommet = nombre de followers du compte. degré sortant d'un sommet = nombre de comptes suivis par ce compte.

Facebook. sommets = comptes Facebook, arcs entre comptes A et B ssi A et B sont ami·es. Graphe non orienté, sans arêtes multiples, sans boucles. ordre = nombre de comptes. degré d'un sommet = nombre d'amis de la personne.

Pages WP. sommets = pages WP, arc entre page x et y ssi x contient un lien vers y. Graphe orienté, sans arêtes multiples, avec boucles (idem qu'au-dessus). ordre = nb de pages WP. degré entrant = nb de pages qui pointent vers cette page. degré sortant = nb de pages vers lesquelles pointe cette page.

Liens hypertextes. sommets = URL (pages web), arcs = liens Graphe orienté, multi-arêtes (il peut y avoir plusieurs liens entre deux pages), avec boucles (une page peut contenir un lien vers elle-même), valué (on étiquette chaque arc avec le texte du lien). ordre = nb de pages web existantes, ou ayant existé (car un lien peut être mort) — voire même davantage, puisqu'il y a des liens erronés, qui n'ont jamais fonctionné. degré sortant = nb de liens dans la page degré entrant = nb de liens qui pointent vers cette page NB : remarquer que la taille du graphe est le nb total de liens sur le web.

Pour écrire des algorithmes sur des graphes, on s'autorise les opérations suivantes.

Pour un graphe non orienté :	Pour un graphe orienté :
— parcourir ses sommets, ses arêtes ; — parcourir les arêtes incidentes et/ou les sommets adjacents d'un sommet donné ; — récupérer les extrémités et le poids d'une arête donnée via une fonction `extremites()` et `poids()`.	— parcourir ses sommets, ses arcs ; — parcourir les arcs sortants ou entrants d'un sommet donné ; — récupérer les extrémités et le poids d'un arc donné via des fonctions `source()`, `dest()`, et `poids()`.

Exercice 2 (Graphe d'Instagram)**.** On va maintenant écrire des algorithmes pour répondre à des questions sur le graphe d'Instagram (ou Twitter, Bluesky, Threads…). Écrire le pseudo-code des fonctions renvoyant les informations suivantes. (NB : elles prennent toutes en entrée le graphe, en plus de leur éventuel paramètre).

1. Nombre de comptes Instagram.
2. Nombre de comptes suivis par un compte donné en entrée.
3. Nombre de *followers* d'un compte donné en entrée.
4. Nombre de comptes ayant plus de 1000 *followers*.
5. Nombre maximal de *followers*.
6. Compte le plus suivi, en supposant qu'il est unique.
7. Idem, mais sans supposer qu'il est unique – il faut donc renvoyer un ensemble de comptes, qui sera un singleton s'il n'y a pas d'égalité.

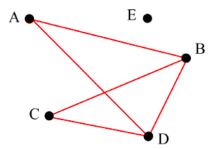
```
fonction nbComptes(G):
  nb = 0
  pour chaque sommet s de G:
    nb += 1
  renvoyer nb


fonction nbComptesSuivis(G, s):
  nb = 0
  pour chaque arc sortant de s:
    nb += 1
  renvoyer nb


fonction nbFollowers(G, s):
  nb = 0
  pour chaque arc entrant de s:
    nb += 1
  renvoyer nb


fonction nbComptes1000Followers(G):
  nb = 0
  pour chaque sommet s de G:
    si nbFollowers(G, s) > 1000:
      nb += 1
  renvoyer nb


fonction nbMaximalFollowers(G):
  max = -1
```

```
    pour chaque sommet s de G:
      nbF = nbFollowers(G, s)
      si nbF > max:
        max = nbF
    renvoyer max


  fonction compteLePlusSuivi(G):
    max = -1
    argmax = nil
    pour chaque sommet s de G:
      nbF = nbFollowers(G, s)
      si nbF > max:
        max = nbF
        argmax = s
    renvoyer argmax

  fonction comptesLesPlusSuivis(G):
    max = -1
    argmax = ensemble vide
    pour chaque sommet s de G:
      nbF = nbFollowers(G, s)
      si nbF > max:
        max = nbF
        argmax = ensemble vide
      si nbF == max:
        ajouter s à argmax
    renvoyer argmax
```

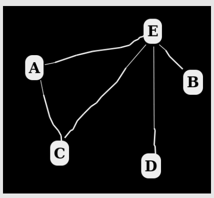
Exercice 3 (Graphe complémentaire)**.** On se place dans le contexte des graphes simples non orientés (donc sans arêtes multiples et sans boucle).

Soit un graphe $G = (S, A)$. Le graphe *complémentaire* de G, noté $\overline{G}$, est le graphe avec les mêmes sommets que G, tel que pour toute paire de sommets distincts s, s', il y a une arête entre s et s' dans $\overline{G}$ si et seulement s'il n'y a pas d'arête entre s et s' dans G.

1. Dessiner le complémentaire du graphe suivant :



2. Soit G un graphe ; que peut-on dire du complémentaire de $\overline{G}$? Le prouver formellement.
3. Que peut-on dire du complémentaire d'un graphe complet à n sommets ?
4. Le complémentaire d'un graphe régulier est-il forcément régulier ? Justifier.

1. 

2. Le complémentaire de $\overline{G}$ est G lui-même. Preuve : l'ensemble de sommets des trois graphes $G, \overline{G}, \overline{\overline{G}}$ est le même ; pour montrer que l'ensemble d'arêtes de $\overline{\overline{G}}$ est le même que G, il suffit de considérer deux sommets quelconques s et s' : il y a une arête entre

eux dans $\overline{\overline{G}}$ si et seulement s'il n'y en a pas dans $\overline{G}$, si et seulement s'il y en a une dans G, CQFD.

3. Le complémentaire d'un graphe complet à n sommets est un *graphe vide* (c'est-à-dire sans arête) à n sommets.
4. Le complémentaire d'un graphe régulier est toujours régulier. En effet, si on note $\mathrm{d}_G(s)$ le degré d'un sommet s dans un graphe $G = (S, A)$ d'ordre n, alors on a $\mathrm{d}_G(s) + \mathrm{d}_{\overline{G}}(s) = n - 1$, car pour chaque sommet distinct de s, soit il y a une arête entre s et s' dans G (et donc pas dans $\overline{G}$), soit il n'y en a pas et il y en a donc une dans $\overline{G}$). Si tous les sommets ont le même degré D dans G, alors pour tout sommet s, on a $\mathrm{d}_{\overline{G}}(s) = n-1-D$, c'est-à-dire que tous les sommets ont le même degré $n-1-D$ dans $\overline{G}$.

Exercice 4 (Livre de recettes)**.** On considère un livre de recettes de cuisine. On considère le graphe des ingrédients de ce livre : les sommets sont des ingrédients, et une arête entre deux ingrédients (distincts) a et b indique combien de recettes utilisent ces deux ingrédients ensemble.

1. De quel type de graphe s'agit-il ?

C'est un graphe non orienté, sans arêtes multiples et sans boucle, mais pondéré.

2. Écrire le pseudo-code des fonctions qui renvoient les informations suivantes. On pourra récupérer le poids d'une arête via la fonction `poids(a)` ou `poids(s, s')`.
   (a) Nombre d'ingrédients avec lesquels un ingrédient donné peut être combiné.
   (b) Ingrédient pour lequel il existe le plus de recettes qui le combinent à un ingrédient donné (en supposant qu'il est unique).
   (c) Ingrédient pouvant être combiné avec le plus d'ingrédients différents (en supposant qu'il est unique).

```
fonction nbIngredientsCombinables(G, s):
  nb = 0
  pour chaque arête a incidente à s:
    nb += 1
  renvoyer nb


fonction ingredientLePlusCombinableAvec(G, s):
  max = -1
  argmax = nil
  pour chaque sommet s' adjacent à s:
    si poids(s, s') > max:
      max = poids(s, s')
      argmax = s'
  renvoyer argmax


fonction ingredientLePlusVersatile(G):
  max = -1
  argmax = nil
  pour chaque sommet s de G:
    nb = nbIngredientsCombinables(G, s)
    si nb > max:
      max = nb
      argmax = s
```

```
renvoyer s
```

Exercice 5 (Voisins de Hamming)**.** On dit que deux entiers sont des *voisins de Hamming* si leur écriture binaire ne diffère que d'un seul bit. Par exemple, 11 et 9 sont des voisins de Hamming (il suffit de changer le bit de poids 2 pour passer de l'un à l'autre), de même que 13 et 29 (bit de poids 16). En revanche, 12 et 9 ne le sont pas (deux bits diffèrent), et 0 et 31 encore moins (5 bits diffèrent) — alors que 0 et 32 sont, eux, voisins.

1. Dessiner le graphe des voisins de Hamming pour les entiers de 0 à 7.

   Voir dessin plus loin, il suffit d'enlever les flèches.

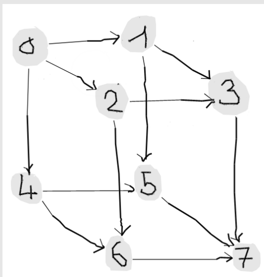
2. De quel type de graphe s'agit-il ?

   Graphe non orienté, simple (sans boucle, sans arête multiple).

3. Quel est le degré minimal et maximal des sommets du graphe ? Quelle propriété du graphe peut-on en déduire ?

   3 et 3. On dit que le graphe est régulier (et plus précisément 3-régulier).

4. Orienter à présent les arêtes, de façon à ce que la source d'un arc soit inférieure à sa destination.

   On peut le dessiner en mettant la face 2367 à l'intérieur de l'autre (c'est un graphe planaire). Je pense qu'on voit moins bien le cube, qui est intéressant : par exemple chaque paire de faces opposées correspond à un bit, et les grandes diagonales du cube relient les paires d'entiers qui sont le XOR bit-à-bit l'un de l'autre. Une autre propriété rigolote est que tous les chemins orientés entre deux sommets x et y ont la même longueur.

5. Comment le fait qu'il y ait un arc de x à y se traduit-il sur l'écriture binaire de x et y ?

   On peut transformer x en y en changeant un bit de 0 à 1.

6. À quoi correspondent les degrés entrant et sortant d'un entier x ? (NB : la réponse attendue n'est correcte que parce qu'on a inclus *tous et seulement* les entiers représentables sur trois bits.)

   Degré entrant = nombre de bits à 1 dans l'écriture sur 3 bits
   Degré sortant = nombre de bits à 0 dans l'écriture sur 3 bits

Correction TD 2 : Degrés, isomorphismes

Exercice 1 (Nombre de sommets et degré)**.**

1. Montrez qu'un graphe simple non orienté a un nombre pair de sommets de degré impair.

Preuve courte : Soit $G = (S, A)$ un graphe simple non orienté, et notons N la somme des degrés de ses sommets, $N = \sum_{s \in S} \mathrm{d}(s)$. La formule de la somme des degrés nous dit que $N = 2 \cdot |A|$, c'est donc un nombre pair. On peut décomposer N en deux morceaux : la contribution des sommets de degré pair, et celle des sommets de degré impair. La contribution des sommets de degré pair à N est forcément paire (c'est une somme de nombres pairs). Donc celle des sommets de degré impair doit également l'être (puisque la somme des deux contributions est un nombre pair). Or la somme d'un nombre impair de nombres impairs ne peut pas être paire. Par l'absurde, il ne peut y avoir qu'un nombre pair de sommets de degré impair.

Preuve détaillée : (On ne s'attend pas forcément à un tel niveau de détail dans ce que vous rendez.)
Soit $G = (S, A)$ un graphe simple non orienté. Notons S_{pair} l'ensemble de ses sommets de degré pair, et S_{impair} l'ensemble de ses sommets de degré impair.
On écrit la formule de la somme des degrés :

$$\sum_{s \in S} \mathrm{d}(s) = 2 \cdot |A|$$

Puisque $S = S_{\text{pair}} \cup S_{\text{impair}}$ et $S_{\text{pair}} \cap S_{\text{impair}} = \emptyset$, et que l'addition est associative et commutative, on peut décomposer la première somme en deux parties, et obtenir l'équation suivante.

$$\sum_{s \in S_{\text{pair}}} \mathrm{d}(s) + \sum_{s \in S_{\text{impair}}} \mathrm{d}(s) = 2 \cdot |A|$$

Pour chaque sommet s, s'il est de degré pair, il existe un entier k_s tel que $\mathrm{d}(s) = 2k_s$, et s'il est de degré impair, il existe un entier k_s tel que $\mathrm{d}(s) = 2k_s + 1$. (NB : les entiers k_s dépendent de s.) On a donc :

$$\begin{aligned}
\sum_{s \in S_{\text{pair}}} (2k_s) + \sum_{s \in S_{\text{impair}}} (2k_s + 1) &= 2 \cdot |A| \\
2(\sum_{s \in S_{\text{pair}}} k_s) + (\sum_{s \in S_{\text{impair}}} (2k_s) + \sum_{s \in S_{\text{impair}}} 1) &= 2 \cdot |A| \\
2(\sum_{s \in S_{\text{pair}}} k_s) + 2(\sum_{s \in S_{\text{impair}}} k_s) + |S_{\text{impair}}| &= 2 \cdot |A|
\end{aligned}$$

On soustrait les deux premiers termes à droite et à gauche, puis on factorise :

$$\begin{aligned}
|S_{\text{impair}}| &= 2 \cdot |A| - 2(\sum_{s \in S_{\text{pair}}} k_s) - 2(\sum_{s \in S_{\text{impair}}} k_s) \\
&= 2 \cdot (|A| - \sum_{s \in S_{\text{pair}}} k_s - \sum_{s \in S_{\text{impair}}} k_s)
\end{aligned}$$

Le nombre de sommets de degrés impair, $|S_{\text{impair}}|$, s'écrit donc sous la forme $2K$ avec K un entier, c'est donc un nombre pair.

2. Est-il possible de relier 15 ordinateurs de sorte que chaque appareil soit relié avec exactement trois autres ? Si oui, construire le graphe, sinon, écrire une démonstration.

Impossible par le résultat précédent. Si c'était possible, le graphe aurait un nombre impair (15) de sommets de degré impair (3).

Exercice 2 (Compléments sur le complémentaire)**.** On se place dans le contexte des graphes simples non orientés.

1. Existe-t-il un graphe G qui soit *égal* à son complémentaire (c'est-à-dire tel que $G = \overline{G}$) ?

C'est le cas de tout graphe d'ordre inférieur ou égal à 1. Ils ont les mêmes sommets que leur complémentaire (par déf) et ont 0 arête, autant que leur complémentaire.

2. Démontrer que pour tout entier $n \geq 2$, il n'existe pas de graphe d'ordre n qui soit égal à son complémentaire.

Dans un graphe G d'ordre $n \geq 2$, on peut choisir deux sommets distincts s et s'. S'il y a une arête entre s et s' dans G, il n'y en a pas dans $\overline{G}$, et réciproquement, donc G ne peut pas être égal à $\overline{G}$.

3. Existe-t-il un graphe G d'ordre 4 qui soit *isomorphe* à son complémentaire $\overline{G}$?

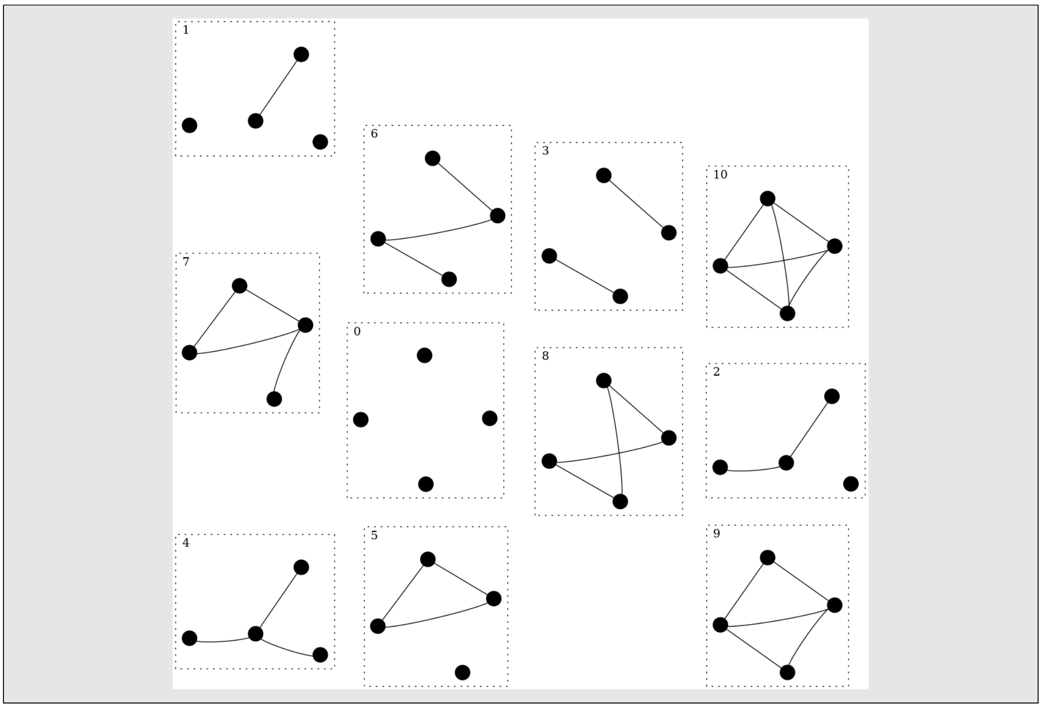
C'est le cas notamment du graphe-chemin P_4, son complémentaire est aussi un P_4 – mais attention, ce n'est pas le même, on le voit bien si on numérote les sommets.

4. Écrire le pseudo-code d'une fonction qui renvoie le complémentaire d'un graphe donné en entrée. On pourra utiliser les fonctions suivantes :
   - créerGraphe() renvoie un nouveau graphe, sans sommet
   - ajouterSommet(G, s) ajoute un sommet s au graphe G (ou ne fait rien si s est déjà dans G)
   - ajouterArête(G, s, s') ajoute une arête entre les sommets existants s et s' dans G (ou ne fait rien si l'arête existe déjà dans G)
   - voisins(G, s, s') renvoie vrai si s et s' sont voisins dans G, et faux sinon.

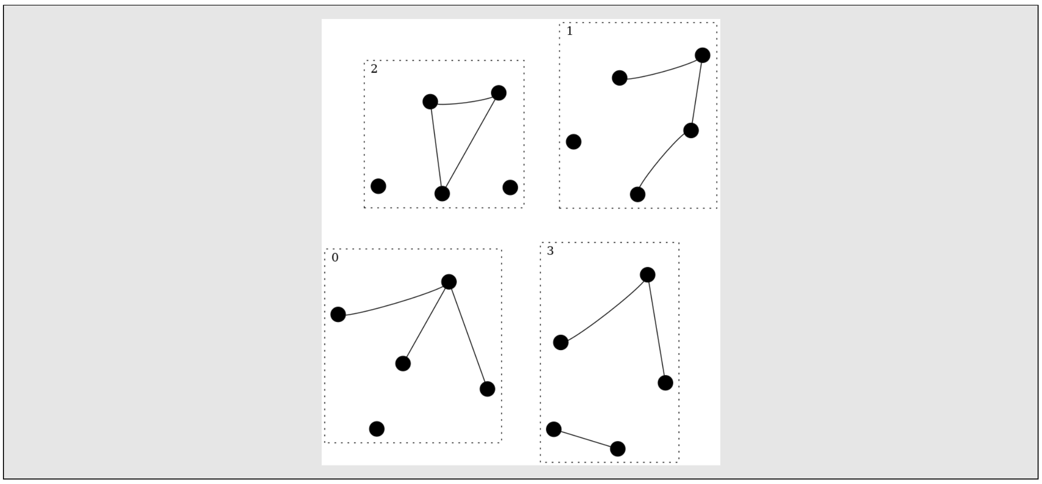
```
fonction complementaire(G):
  G' = créerGraphe()
  pour chaque sommet s de G:
    ajouterSommet(G', s)

  pour chaque sommet s de G:
    pour chaque sommet s' de G:
      si s != s' et non voisins(G, s, s'):
        ajouterArête(G', s, s')
  renvoyer G'
```

Exercice 3 (Isomorphismes)**.** On considère les familles de graphes suivantes, c'est-à-dire des ensembles de graphes qui respectent certaines contraintes :

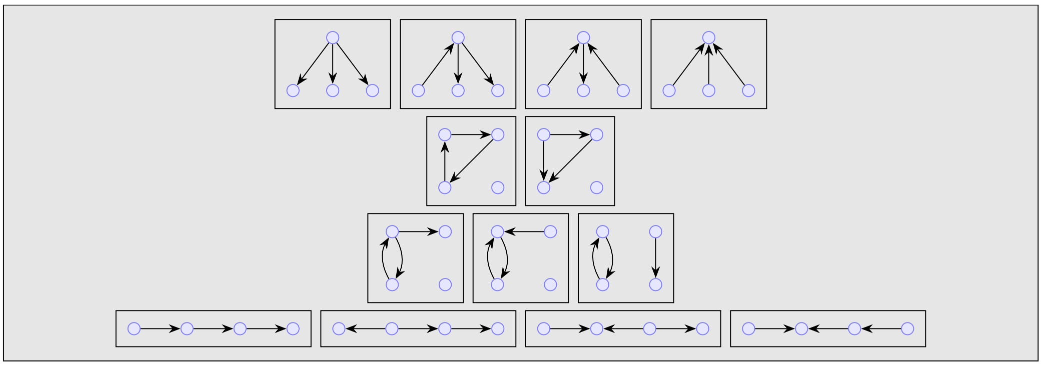
1. Les graphes simples non orientés d'ordre 4.



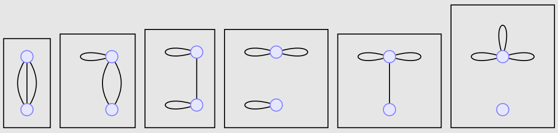
2. Les graphes simples non orientés d'ordre 5 et de taille 3.



3. Les graphes simples orientés d'ordre 4 ayant exactement trois arcs.

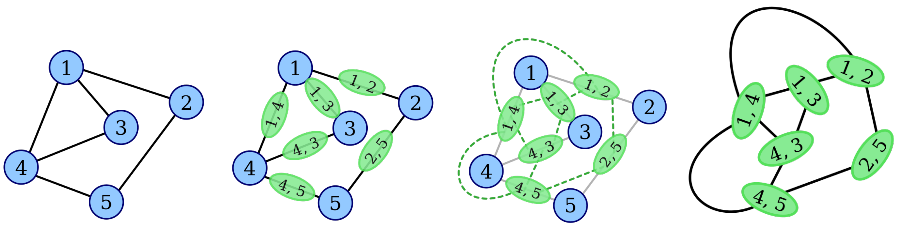


4. Les multigraphes d'ordre 2 et de taille 3.

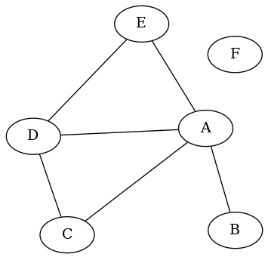


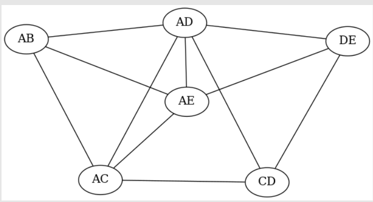
Pour chaque famille, dessiner tous les graphes de cette famille *à isomorphisme près*. Autrement dit, dessiner le plus possible de graphes de la famille, sans dessiner un graphe isomorphe à un autre déjà dessiné. Normalement, tous les graphes de la famille doivent être isomorphes à l'un de vos graphes.

Exercice 4 (Graphe adjoint)**.** On se place dans le contexte des graphes simples non orientés. Le graphe *adjoint* (*line graph*) d'un graphe $G = (S, A)$ est le graphe $L(G) = (S', A')$ où $S' = A$ et $x, y \in A' \iff x \cap y \neq \emptyset$. Autrement dit, les sommets de l'adjoint correspondent aux arêtes de G, et deux sommets sont adjacents ssi les arêtes correspondantes dans G sont incidentes à un sommet commun. Exemple de construction d'un graphe adjoint (piqué sur Wikipédia) :



1. Construire le graphe adjoint du graphe suivant :





2. Quel est le graphe adjoint de K_1 ?

Le graphe d'ordre zéro (K_0), c'est-à-dire le graphe sans sommet.

3. Trouver un graphe G tel que $L(G)$ soit isomorphe à $\overline{K}_4$ (le graphe d'ordre 4 sans arête).

Il faut qu'il y ait quatre arêtes dans G et qu'elles ne partagent pas de sommets. Le graphe consistant en quatre copies disjointes de K_2 fonctionne (on peut aussi y ajouter autant de sommets isolés qu'on veut).

4. Trouver un graphe G tel que $L(G) = \overline{G}$.

C'est le cas de K_0, dont l'adjoint et le complémentaire sont K_0 (c'est donc bien une égalité, pas un isomorphisme !)

5. Trouver un graphe G d'ordre 3 qui soit isomorphe à $L(G)$. Idem pour l'ordre 4.

C'est le cas pour $K_3 = C_3$ et pour C_4 (et pour C_n pour tout $n \geq 3$)

6. Trouver un graphe G d'ordre 4 qui soit isomorphe à $L(L(G))$.

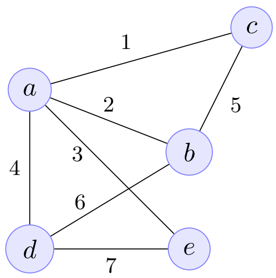
On a vu que que C_4 est isomorphe à $L(C_4)$, il l'est donc aussi à $L(L(C_4))$!

7. Écrire le pseudo-code d'une fonction qui construit le graphe adjoint d'un graphe donné en entrée. (On pourra s'aider d'un dictionnaire comme en Python.)

```
fonction grapheAdjoint(G):
  G' = créerGraphe()
  d = nouveau dictionnaire
  i = 0
  pour chaque arête a de G:
    ajouterSommet(G', i)
    d[a] = i
    i += 1
  pour chaque sommet s de G:
    pour chaque arête a incidente à s:
      pour chaque arête a' incidente à s:
        si a != a':
          ajouterArête(G', d[a], d[a'])
  renvoyer G'
```

Correction TD 3 : Sous-graphes, cliques, stables

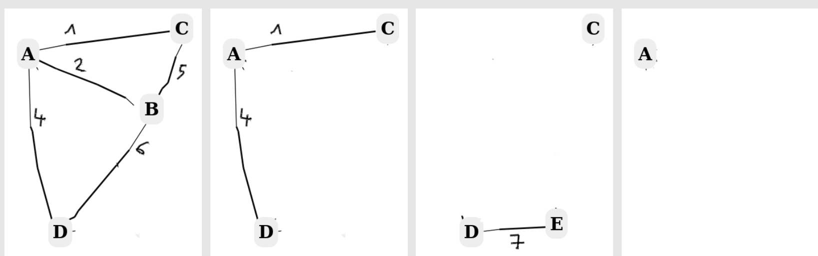
Exercice 1 (Sous-graphes)**.** On considère le graphe simple non orienté G suivant, dont on a numéroté les arêtes (ce ne sont pas des poids) :



1. Pour chacun des ensembles de sommets et d'arêtes suivants, indiquer s'ils constituent un sous-graphe de G.
   (a) $S = \{a, b, d, e\}$, $A = \{2, 6, 7\}$
   (b) $S = \{a, b, d, e\}$, $A = \{3, 4, 5\}$
   (c) $S = \{a, b, d, e\}$, $A = \{\{a, d\}, \{a, e\}, \{a, b\}\}$
   (d) $S = \{a, b, d, e\}$, $A = \{\{a, d\}, \{d, e\}, \{b, e\}\}$

> (a) oui
> (b) non (5 est incidente à c qui n'est pas dans les sommets)
> (c) oui
> (d) non ($\{b, e\}$ n'est pas une arête dans G)

2. Dessiner les sous-graphes de G induits par les ensembles de sommets suivants :
   (a) $\{a, b, c, d\}$
   (b) $\{a, c, d\}$
   (c) $\{c, d, e\}$
   (d) $\{a\}$



3. Pour chacun des ensembles de sommets et d'arêtes suivants, indiquer s'ils constituent un sous-graphe induit de G.
   (a) $S = \{a, b, e\}$, $A = \{2, 3, 5\}$
   (b) $S = \{a, b, d\}$, $A = \{2, 6\}$
   (c) $S = \{c, d\}$, $A = \emptyset$
   (d) $S = \{c, d, a\}$, $A = \emptyset$

(a) non, ce n'est même pas un sous-graphe (5 est incidente à c qui n'est pas dans les sommets)

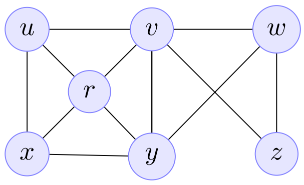
(b) non, pour qu'il soit induit il faudrait l'arête 4

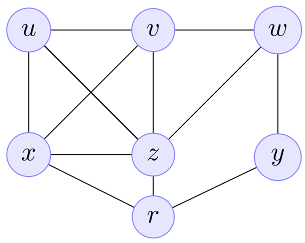
(c) oui

(d) non, il manque les arêtes 1 et 4

Exercice 2 (Cliques et stables)**.** Répondez aux questions suivantes pour chacun des deux graphes suivants.

- Lister toutes les cliques maximales du graphe.
- Donner la taille d'une clique maximum du graphe.
- Lister tous les stables maximaux du graphe.
- Donner la taille d'un stable maximum du graphe.





(Notation simplifiée pour aller vite, à ne pas reproduire)

1. Cliques maximales : uvr, uxr, xry, rvy, vwz, vyw. Taille d'une clique maximum : 3. Stables maximaux : uyz, uw, xv, xw, xz, rw, rz. Taille d'un stable maximum : 3.
2. Cliques maximales : uvxz, xzr, vwz, wy, ry. Taille d'une clique maximum : 4. Stables maximaux : uwr, uy, vr, vy, wx, xy, zy. Taille d'un stable maximum : 3.

Exercice 3. On se place dans le contexte des graphes simples non orientés. Dans cet exercice, on appelle *voisinage d'un sommet* s l'ensemble contenant s et tous ses voisins ; par exemple, dans le graphe de l'exercice 1, le voisinage de d est $\{a, b, d, e\}$.

Prouver que si le voisinage d'un sommet s est une clique, alors c'est une clique maximale.

Si c'est une clique qui n'est pas maximale, ça signifie qu'il y a un autre sommet s' qui n'est pas dans le voisinage de s mais est pourtant voisin de tous les sommets du voisinage de s, donc aussi de s, ce qui est contradictoire.

Exercice 4 (Test de cliques)**.** On se place dans le contexte des graphes simples non orientés. On suppose qu'on dispose des fonctions `taille (G)` et `ordre(G)`, ainsi que d'une fonction `complementaire (G)`. Comme d'habitude, on ne se préoccupera pas de l'efficacité.

1. Écrire une fonction `induit (G, C)` qui prend en entrée un graphe G et un ensemble de sommets C, et qui renvoie $G|_C$, le sous-graphe de G induit par C.
2. Écrire une fonction `estUneClique(G, C)` qui prend en entrée un graphe G et un ensemble de sommets C, et qui renvoie vrai si et seulement si C est une clique dans G.
3. Écrire une fonction `estUneCliqueMaximale(G, C)` qui prend en entrée un graphe G et un ensemble de sommets C, et qui renvoie vrai si et seulement si C est une clique maximale dans G.
4. Écrire une fonction `estUnStableMaximal(G, C)` qui prend en entrée un graphe G et un ensemble de sommets C, et qui renvoie vrai si et seulement si C est un stable maximal dans G.

```
fonction induit(G, C):
  G' = créerGraphe()
  pour chaque élément s de C:
    ajouterSommet(G', s)
  pour chaque arête a de G:
    si les deux extrémités de a sont dans C:
      ajouterArête(G', a)
  renvoyer G'

fonction estUneClique(G, C):
  G' = induit(G, C)
  n = ordre(G')
  si taille(G') = n * (n-1) / 2:
    renvoyer vrai
  sinon
    renvoyer faux

fonction estUneCliqueMaximale(G, C):
  si non estUneClique(G, C):
    renvoyer faux
  pour chaque sommet s de G qui n'est pas dans C:
    C' = C union {s}
    si estUneClique(G, C'):
      renvoyer faux
  renvoyer vrai

fonction estUnStableMaximal(G, C):
  renvoyer estUneCliqueMaximale(complementaire(G), C)
```

Exercice 5 (Degrés tous différents)**.** Prouver que tout graphe simple non orienté d'ordre supérieur ou égal à 2 possède au moins deux sommets de même degré.

Considérons un graphe d'ordre n. Le degré de chaque sommet est forcément compris entre 0 et $n-1$ (inclus). Cependant, il ne peut pas y avoir à la fois un sommet de degré 0 et un sommet de degré $n-1$: en effet, si un sommet est de degré 0, il n'est relié à aucun autre, donc le degré maximal de tous les autres est $n-2$. Par conséquent, les degrés sont soit tous entre 0 et $n-2$, soit tous entre 1 et $n-1$. Dans les deux cas, il n'y a que $n-1$ valeurs différentes possibles pour le degré. Il y a donc plus de sommets que de degrés possibles ($n > n-1$), donc il y a au moins deux sommets de même degré (principe des tiroirs).

Exercice 6. On se place dans le contexte des graphes simples non orientés. On considère la proposition suivante :

> Pour tout sous-graphe H d'un graphe $G = (S, A)$, il existe un ensemble de sommets $S' \subseteq S$ tel que $H = G|_{S'}$.

Démontrer soit que cette proposition est vraie, soit qu'elle est fausse.

Si H est le graphe induit par S' dans G, l'ensemble des sommets de H est forcément S'. La proposition revient donc à dire que tout sous-graphe est un sous-graphe induit, ce qui est faux (par exemple dans l'exercice 1 le sous-graphe $(\{a, b, c\}, \{1, 2\})$ n'est pas induit).

Exercice 7 (Test de clique maximum)**.** On prend la suite de l'exercice 4.

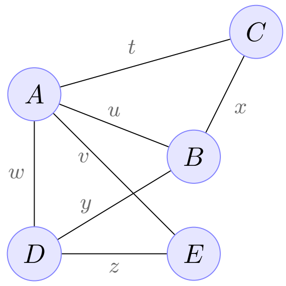
1. Écrire le pseudo-code d'une fonction sousEnsembles(E) qui renvoie l'ensemble de tous les sous-ensembles d'un ensemble E.

2. En utilisant cette fonction, écrire le pseudo-code d'une fonction qui renvoie la taille d'une clique maximum d'un graphe G.

```
fonction sousEnsembles(E):
  res = ensemble vide
  ajouter l'ensemble vide dans res
  pour chaque élément e de E:
    nouveaux = ensemble vide
    pour chaque élément r de res:
      ajouter r union {e} dans nouveaux
    res = res union nouveaux
  renvoyer res

fonction tailleCliqueMaximum(G):
  max = −1
  pour chaque élément C de sousEnsembles(sommets(G)):
    c = |C|
    si c > max:
      si estUneClique(G, C):
        max = c
  renvoyer max
```

Correction TD 4 : Chemins et connexité

Exercice 1 (Chemins)**.** On considère le graphe simple non orienté G suivant, dont on a nommé les arêtes :



1. Pour chacune des suites de sommets et d'arêtes suivantes , indiquer si elle est un chemin dans G, et si oui, donner sa longueur.

   (a) $(E, v, A, u, B, u, A, w, D)$
   (b) (A, w, z, E, v, A)
   (c) (A, u, B, v, E)
   (d) (t, C, x, B)
   (e) (D)

   > (a) oui, de longueur 4
   > (b) non, on a deux arêtes consécutives sans sommet (w et z)
   > (c) non, v n'est pas incidente à B
   > (d) non, un chemin commence forcément par un sommet or t est une arête
   > (e) oui, de longueur 0

2. Pour chacune des suites de sommets suivantes, indiquer si elle correspond à un chemin dans G, et si oui, donner le chemin et sa longueur.

   (a) (A, B, D, E)
   (b) (A, E, B, C)
   (c) (D, B, A, A, C)
   (d) (D, B, A, C, A)
   (e) (C, C)
   (f) (E)
   (g) $()$ (la suite de longueur 0)

   > (a) oui, (A, u, B, y, D, z, E), longueur 3
   > (b) non, il n'y a pas d'arête entre E et B
   > (c) non, il n'y a pas d'arête entre A et lui-même
   > (d) oui, $(D, y, B, u, A, t, C, t, A)$, longueur 4
   > (e) non, il n'y a pas d'arête entre C et lui-même
   > (f) oui, (E), longueur 0
   > (g) non, un chemin commence forcément par un sommet.

3. Quelle est la distance entre les couples de sommets suivants ? Donner un plus court chemin à chaque fois.

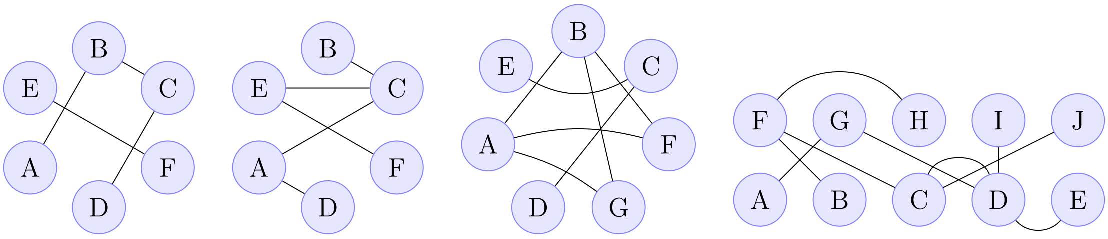
(a) A et A (b) B et E (c) C et D (d) C et E

0 : (A)
2 : (B, u, A, v, E) ou (B, y, D, z, E)
2 : (C, x, B, y, D) ou (C, t, A, w, D)
2 : (C, t, A, v, E)

4. Quel est le plus long chemin de B à D qui ne passe jamais deux fois par une même arête ?

NB : le chemin en question passe plusieurs fois par un même sommet (y compris B et D), mais ce n'est pas interdit par l'énoncé.
Le chemin est $(B, x, C, t, A, u, B, y, D, w, A, v, E, z, D)$, ce qui correspond à la suite de sommets (B, C, A, B, D, A, E, D). Il est clair qu'il ne peut pas y avoir de plus long chemin sans repasser une deuxième fois par une arête, puisque le chemin passe par toutes les arêtes.

Exercice 2 (Connexité)**.** Pour chacun des graphes suivants, indiquez s'il est connexe ou non, et lister ses composantes connexes (sous la forme d'ensembles de sommets).



1. non (pas de chemin notamment entre A et E). 2 composantes ABCD et EF.
2. oui. 1 composante ABCDEF.
3. non (pas de chemin notamment entre A et E). 2 composantes ABFG et CDE.
4. oui. 1 composante ABCDEFGHIJ.

Exercice 3 (Algos)**.** Écrire un pseudo-code pour les fonctions suivantes.

1. existeCheminLongueurN(G, s1, s2, n), qui prend en entrée un graphe simple non orienté, deux sommets, et un entier naturel n, et qui renvoie vrai ssi il existe un chemin de longueur n entre s1 à s2. Comme d'habitude, on ne se préoccupe pas de l'efficacité, on cherche à écrire des algorithmes conceptuellement simples.
2. distance(G, s1, s2), qui renvoie la distance entre deux sommets (NB : la distance à un sommet non atteignable est $+\infty$). (On suppose qu'on a une fonction taille (G).)
3. estConnexe(G) qui renvoie vrai ssi G est connexe.

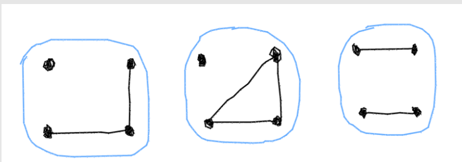
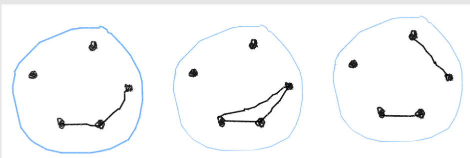
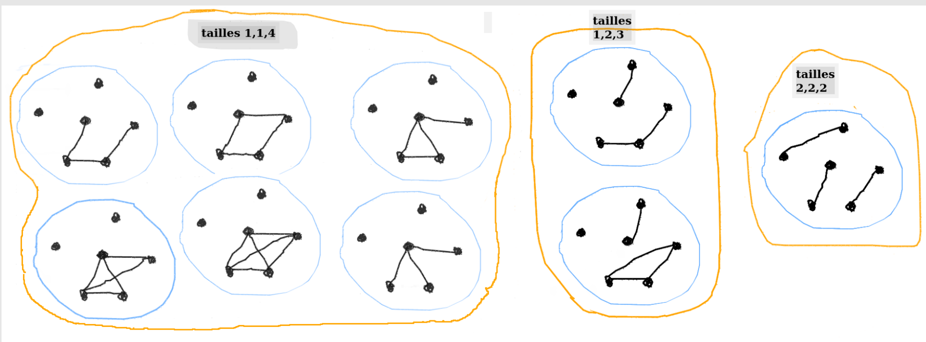
```
fonction existeCheminLongueurN(G, s1, s2, n):
  si n = 0:
    renvoyer (s1 = s2)
  pour chaque voisin v de s1:
    si existeCheminLongueurN(G, v, s2, n-1):
      renvoyer vrai
  renvoyer faux


fonction distance(G, s1, s2):
```

```
  pour i variant de 0 à taille(G):      # NB: on s'arrête à taille(G)
                                        # car un plus court chemin
                                        # ne contient jamais
                                        # d'arête en double
    si existeCheminLongueurN(G, s1, s2, i):
      renvoyer i
  renvoyer +oo


fonction estConnexe(G):
  si ordre(G) = 0:
    renvoyer faux
  pour chaque sommet s de G:
    pour chaque sommet s' de G:
      si distance(G, s, s') = +oo:
        renvoyer faux
  renvoyer vrai
```

Exercice 4 (Composantes connexes)**.** On considère les familles de graphes suivantes, c'est-à-dire des ensembles de graphes qui respectent certaines contraintes :

1. Les graphes simples non orientés d'ordre 4 ayant exactement deux composantes connexes.
2. Les graphes simples non orientés d'ordre 5 ayant exactement trois composantes connexes.
3. Les graphes simples non orientés d'ordre 6 ayant exactement trois composantes connexes.

Pour chaque famille, dessiner tous les graphes de cette famille *à isomorphisme près*. Autrement dit, dessiner le plus possible de graphes de la famille, sans dessiner un graphe isomorphe à un autre déjà dessiné. Normalement, tous les graphes de la famille doivent être isomorphes à l'un de vos graphes.

1. 
2. 
3. 

Exercice 5 (Graphes de clusters)**.** On appelle *graphe de clusters* un graphe (simple non orienté) dont toutes les composantes connexes sont des graphes complets.

1. Montrer le lemme suivant :

> Un graphe (simple non orienté) est un graphe de clusters si et seulement si la distance entre toute paire de sommets distincts s et s' est soit 1, soit $+\infty$.

($\Rightarrow$) Soient deux sommets distincts s et s' d'un graphe de clusters. S'ils appartiennent à une même composante connexe, alors il existe une arête entre s et s' puisque chaque composante connexe est complète ; ils sont donc à distance 1. S'ils n'appartiennent pas à la même composante connexe, par définition s' n'est pas atteignable depuis s, donc la distance entre les deux est infinie.

($\Leftarrow$) Soit un graphe dont la distance entre toute paire de sommets distincts est soit 1 soit $+\infty$. Soit une composante connexe G' du graphe ; il faut montrer que G' est complet. Considérons s et s' deux sommets distincts de G'. Comme G' est connexe, s' est atteignable depuis s, donc $d(s, s') < +\infty$, donc par hypothèse $d(s, s') = 1$, ce qui signifie que s et s' sont voisins. C'est vrai pour n'importe quelle paire de sommets distincts de G', qui est donc complet.

2. Soit $\mathcal{R}$ une relation symétrique sur un ensemble S. Dans cet exercice, on appelle *graphe de* $\mathcal{R}$ le graphe simple non orienté $G = (S, A)$ tel que pour toute paire de sommets distincts $\{s, s'\}$, on a $\{s, s'\} \in A \iff s \mathrel{\mathcal{R}} s'$. Montrer que le graphe d'une relation symétrique et transitive (par exemple le graphe des frères et sœurs dont on a parlé en cours) est un graphe de clusters.

Supposons $\mathcal{R}$ symétrique et transitive, notons G le graphe de $\mathcal{R}$, et considérons deux sommets distincts s et s' de G. On va montrer par l'absurde que la distance entre s et s' ne peut être que 1 ou $+\infty$.

Soit $k \in \mathbb{N}$ tel que $k > 1$. Supposons que $d(s, s') = k$. Cela signifie qu'il existe un chemin de longueur k de s à s', et qu'il n'y en a pas de plus court. Soit $(s, a_1, s_1, a_2, s_2, a_3, \ldots, s_{k-1}, a_k, s')$ un tel chemin, où $a_1, \ldots, a_k$ sont des arêtes et $s_1, \ldots, s_{k-1}$ des sommets intermédiaires. En regardant les premiers sommets du chemin, on remarque que s et s_1 sont voisins, ainsi que s_1 et s_2. Par conséquent, puisque G est le graphe de $\mathcal{R}$, on a $s \mathrel{\mathcal{R}} s_1$ et $s_1 \mathrel{\mathcal{R}} s_2$, ce qui implique que $s \mathrel{\mathcal{R}} s_2$ puisque $\mathcal{R}$ est transitive ; il y a donc une arête a entre s et s_2 puisque G est le graphe de $\mathcal{R}$. Mais alors $(s, a, s_2, a_3, \ldots, s_{k-1}, a_k, s')$ est un chemin de s à s' de longueur $k - 1$, ce qui est absurde puisqu'on a supposé que $d(s, s') = k$.

La distance entre s et s' ne peut donc qu'être 1 ou $+\infty$, donc G est un graphe de clusters (par le lemme démontré à la question précédente).

3. Soit un graphe de clusters d'ordre n avec c composantes, quelle est la taille de ses stables maximaux ?

Les sommets d'un stable dans un graphe de clusters ne peuvent pas venir d'une même composante connexe ; par ailleurs si on ajoute à un stable un sommet d'une composante connexe dont le stable ne contient aucun sommet, on obtient forcément un stable. Les stables maximaux ont donc tous exactement c sommets.

4. Supposons maintenant que chaque composante est d'ordre n_i (pour $1 \leq i \leq c$) ; on a donc $\sum_{i=1}^{c} n_i = n$. Quel est le nombre de stables maximaux (pour l'inclusion) ?

Avec le même raisonnement, pour faire un stable maximal on peut choisir n'importe quel sommet pour chacune des composantes. On a donc $\prod_{i=1}^{c} n_i$ possibilités.

5. Montrer qu'un graphe est un graphe de clusters si et seulement si aucun de ses sous-graphes induits n'est isomorphe à P_3 (graphe chemin de longueur 3).

(⇒) Soit G un graphe de clusters, et soit G' un graphe induit de G. Si G' n'est pas d'ordre 3, il ne peut pas être isomorphe à P_3 ; supposons donc qu'il est d'ordre 3. Il y a deux possibilités :

— soit les trois sommets appartiennent à une même composante connexe, auquel cas G' est isomorphe à K_3 et donc pas à P_3 ;

— sinon, au moins deux sommets appartiennent à des composantes connexes distinctes, auquel cas G' ne peut pas être connexe, et donc pas isomorphe à P_3.

(⇐) On va montrer la contraposée, c'est-à-dire qu'un graphe qui n'est pas un graphe de clusters a forcément un sous-graphe induit isomorphe à P_3. Soit G un graphe qui n'est pas un graphe de clusters. Par le lemme prouvé à la première question, il existe deux sommets distincts s et s' qui sont à distance $1 < k < +\infty$. Considérons un plus court chemin de s à s', et notons s_1 le deuxième sommet le long de ce chemin (qui est distinct de s et de s' puisque le chemin est de longueur $k > 1$) et s_2 le troisième sommet (qui est distinct de s et de s_1, mais qui peut être égal à s'). Le sous-graphe induit par l'ensemble de trois sommets distincts $\{s, s_1, s_2\}$ a des arêtes $\{s, s_1\}$ et $\{s_1, s_2\}$ par construction, mais pas d'arête $\{s, s_2\}$ (car sinon il existerait un chemin de longueur $k - 1$ entre s et s'). Il est donc isomorphe à P_3.

Exercice 6 (Suite de l'exercice 3)**.** Dans cet exercice, il devient nécessaire d'écrire des algos un peu plus subtils. L'esprit est cependant toujours de rester le plus simple possible, pas d'optimiser l'efficacité. (On reste dans le contexte des graphes simples non orientés.)

1. Écrire une fonction qui renvoie la longueur d'un plus long chemin entre deux sommets qui ne passe pas deux fois par une même arête (ou faux s'il n'y a pas de chemin). On pourra écrire d'autres fonctions, et il n'est pas forcément pertinent de réutiliser les résultats de l'exercice 3.
2. Écrire une fonction qui renvoie le nombre de composantes connexes. On pourra utiliser une fonction induit(G, E) qui renvoie le sous-graphe de G induit par l'ensemble de sommets E.

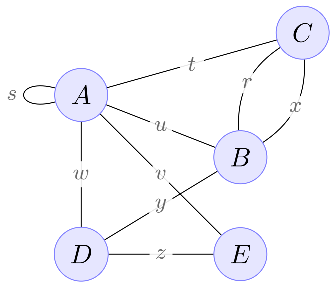
```
fonction existeCheminElemLongueurN(G, s1, s2, n, l):
  pour chaque voisin v de s1:
    si {s1,v} n'est pas dans l:
      si existeCheminElemLongueurN(G, v, s2, n-1, l union {{s1,v}}):
        renvoyer vrai
  renvoyer faux

fonction plusLongCheminElemL(G, s1, s2):
  pour i variant de taille(G) à 0:
    si existeCheminElemLongueurN(G, s1, s2, i, {}):
      renvoyer i
  renvoyer faux

fonction nbComposantesConnexes(G):
  si ordre(G) = 0:
    renvoyer 0
  choisir un sommet s de G
  sommetsNonAtteignables = ensemble vide
  pour chaque sommet s' de G:
    si distance(G, s, s') = +oo:
      ajouter s' dans sommetsNonAtteignables
```

```
renvoyer 1 + nbComposantesConnexes(induit(G, sommetsNonAtteignables))
```

Université de Caen | Théorie des graphes
L3 informatique | Correction TD 5 : Distances, propriétés des chemins

Correction TD 5 : Distances, propriétés des chemins

Exercice 1. On considère le multigraphe non orienté G suivant, dont on a nommé les arêtes :



1. Pour chacun des chemins suivants, indiquer s'il est ouvert ou fermé, s'il est simple, s'il a un croisement, s'il est élémentaire, et si c'est un cycle.

   (a) $(A, u, B, y, D, w, A, v, E)$
   (b) $(C, t, A, u, B, y, D, z, E)$
   (c) $(A, v, E, z, D, y, B, u, A)$
   (d) (B, y, D, y, B)
   (e) (B, r, C, x, B) :
   (f) (A, s, A)
   (g) (B, u, A, s, A) :
   (h) (A)

(a) $(A, u, B, y, D, w, A, v, E)$: chemin ouvert puisque $A \neq E$, simple, avec croisement (A est incident aux 3 arêtes u, w, v) donc non élémentaire, pas un cycle. puisque l'on passe deux fois par a, pas un cycle.

(b) $(C, t, A, u, B, y, D, z, E)$: chemin ouvert puisque $C \neq E$, simple, sans croisement (tous les sommets sont distincts), élémentaire, pas un cycle.

(c) $(A, v, E, z, D, y, B, u, A)$: chemin fermé, simple, sans croisement (tous les sommets sont distincts sauf le premier et le dernier), élémentaire, cycle.

(d) (B, y, D, y, B) : chemin fermé, non simple, sans croisement (tous les sommets sont distincts sauf le premier et le dernier), non élémentaire (car non simple), pas un cycle.

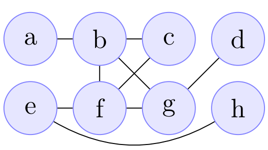
(e) (B, r, C, x, B) : chemin fermé, simple, sans croisement (tous les sommets sont distincts sauf le premier et le dernier), élémentaire , cycle.

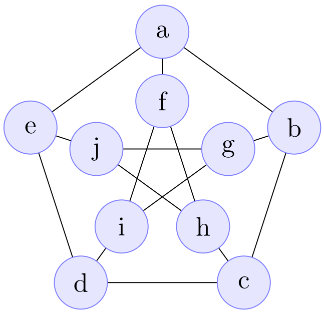
(f) (A, s, A) : chemin fermé, simple, sans croisement, élémentaire, cycle.

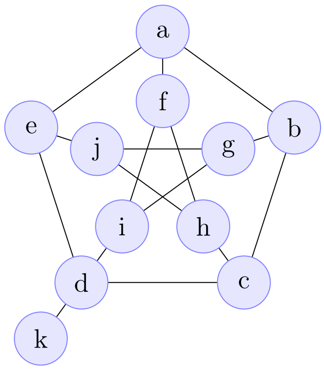
(g) (B, u, A, s, A) : chemin ouvert, simple, avec croisement (A est extrémité de u et deux fois extrémité de s), non élémentaire, pas un cycle.

(h) (A) : chemin fermé (le premier et le dernier sommet sont identiques), simple, sans croisement, élémentaire, pas un cycle (car chemin vide).

Exercice 2. Pour chacun des graphes suivants, déterminer le diamètre, le rayon, et les sommets centraux.

1. 

2. 

3. 

Il suffit de lister les excentricités. Pour le 2e il y a bcp de symétries donc c'est très rapide. Le 3e est facile à traiter à partir du 2e.

1. diamètre 4, rayon 2, sommet central $\{f\}$
2. diamètre = rayon = 2, tous les sommets sont centraux.
3. diamètre 3, rayon 2, sommets centraux $\{c, d, e, i\}$

Exercice 3. On se place dans le contexte des multigraphes non orientés.

1. On considère l'énoncé suivant : « Tout chemin ouvert sans croisement est élémentaire. »Démontrer qu'il est vrai ou démontrer qu'il est faux.

   Nous allons prouver que l'énoncé est vrai. Tout d'abord, remarquons qu'il suffit de montrer que tout chemin ouvert sans croisement est simple (puisque par définition, un chemin sans croisement qui est simple est élémentaire). Pour cela, on va montrer la contraposée, c'est-à-dire que "tout chemin ouvert non simple a forcément (au moins) un croisement" : cet énoncé est équivalent mais plus direct à démontrer.

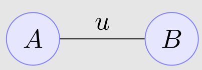
   Soit G un multigraphe non orienté, et soit $C = (s_0, a_1, s_1, \ldots, s_{n-1}, a_n, s_n)$ un chemin ouvert de G qui n'est pas simple. On va montrer qu'il contient (au moins) un croisement.

   Comme C n'est pas simple, il contient une arête qui se répète, c'est-à-dire qu'on peut trouver i, j tels que $1 \leq i < j \leq n$ et $a_i = a_j$.

   Notons $s = s_{i-1}$ le sommet qui précède a_i dans C. Puisque $a_i = a_j$, s est aussi extrémité de a_j. Deux cas sont donc possibles : s est présent soit à l'indice $j-1$ (juste avant a_j), soit à l'indice j (juste après a_j). Puisqu'on a supposé $i < j$, ces deux indices sont différents de $i-1$, donc dans les deux cas le sommet s est répété dans C.

   Cependant, cela ne suffit pas à conclure que s est un croisement : on pourrait imaginer que s soit aux deux extrémités (on aurait $i = 1$ et $j = n$). Mais cela n'est pas possible puisque C est un chemin ouvert. Ainsi, au moins une des deux instances de s est un sommet interne, donc s est un croisement, CQFD.

2. On considère l'énoncé suivant : « Tout chemin sans croisement est élémentaire. »Démontrer qu'il est vrai ou démontrer qu'il est faux.

   L'énoncé est faux. Pour le démontrer, considérons le graphe suivant : 

   Le chemin $c = (A, u, B, u, A)$ dans ce graphe est sans croisement (A n'apparaît pas comme sommet interne) mais n'est pas simple.

Exercice 4. Dans le 2e graphe de l'exercice 2 (qui s'appelle *graphe de Petersen*), trouver :

1. un chemin élémentaire de longueur 9 ;
2. des cycles de longueur 5, 6, 8 et 9.

On note les chemins comme des suites de sommets (non ambigu car c'est un graphe simple).
- $(a, b, c, d, e, j, g, i, f, h)$ est élémentaire de longueur 9.
- cycle de lg 5 : (a, b, g, j, e, a) par ex
- cycle de lg 6 : (a, b, c, h, j, e, a) par ex
- cycle de lg 8 : $(a, b, c, h, j, g, i, f, a)$ par ex
- cycle de lg 9 : $(a, b, c, d, i, f, g, j, e, a)$ par ex

Exercice 5. Soit G un multigraphe non orienté.

1. Montrer que s'il existe un chemin de s à s' dans G, alors il existe un chemin élémentaire de s à s' dans G.

   Pour être vraiment formel on va le montrer par récurrence sur la longueur du chemin. Soit $\mathcal{P}_n$ la proposition suivante : s'il existe un chemin de longueur n de s à s', alors il existe un chemin élémentaire de s à s'.
   - $\mathcal{P}_0$ est vraie car les chemins vides sont élémentaires.
   - Soit $n > 0$; on suppose que $\mathcal{P}_k$ est vraie pour tout $k < n$, et on veut montrer que $\mathcal{P}_n$ est vraie. Supposons donc qu'il existe un chemin de longueur n de s à s', et notons $(s_0, s_1, \ldots, s_n)$ la suite de sommets correspondants (on a $s_0 = s$ et $s_n = s'$). Si le chemin est élémentaire, on a directement le résultat. S'il n'est pas élémentaire, c'est qu'il répète une arête ou contient un croisement. Dans les deux cas, il existe i et j tels que $0 \leq i < j \leq n$ et $s_i = s_j$. Mais alors la partie du chemin entre s_i à s_j est un chemin fermé, qui n'est pas nécessaire pour atteindre s' depuis s : $(s_0, s_1, \ldots, s_{i-1}, s_i, s_{j+1}, \ldots, s_n)$ correspond à un chemin de s à s' de longueur $k < n$ (il a au moins un sommet de moins que le chemin initial). En appliquant l'hypothèse de récurrence $\mathcal{P}_k$, on déduit qu'il existe un chemin élémentaire de s à s'.
   - par récurrence $\mathcal{P}_n$ est donc vraie pour tout $n \in \mathbb{N}$.

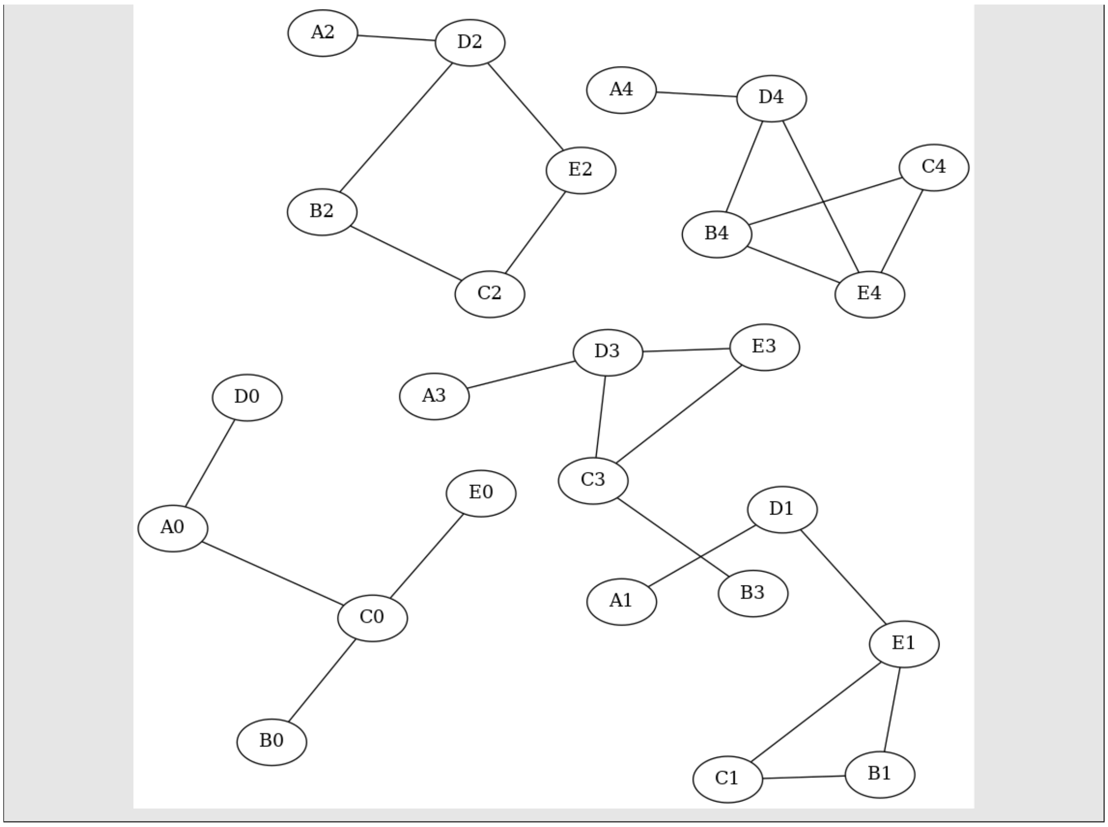
2. Montrer que si $d(s, s') = 3$, tout chemin de longueur 3 de s à s' est élémentaire.

   Soit s et s' tels que $d(s, s') = 3$. Soit un chemin de longueur 3 de s à s'.

   Supposons qu'il n'est pas élémentaire. Ça signifie qu'il répète une arête ou contient un croisement. Dans les deux cas, un sommet est rencontré au moins 2 fois ; si on enlève du chemin les arêtes entre les deux occurrences du sommet répété (comme à la question précédente), on a toujours un chemin de s à s', mais strictement plus court que 3, ce qui contredit le fait que s et s' sont à distance 3. Le chemin est donc élémentaire.

Exercice 6. Dessiner tous les graphes simples non orientés (à isomorphisme près) de rayon 2 et de diamètre 3 d'ordre inférieur ou égal à 5.

Il n'y en a pas d'ordre inférieur ou égal à 3. D'ordre 4, il n'y a que P_4. Ceux d'ordre 5 sont là (ne pas faire attention au fait qu'ils se croisent un peu. . .) :



Exercice 7. Soit G un multigraphe non orienté connexe (donc non nul). Montrer que la fonction de distance d satisfait l'inégalité triangulaire suivante :

$$\forall x, y, z \in S_G : d(x, z) \leq d(x, y) + d(y, z)$$

Soient x, y et z trois sommets de G (pas forcément distincts !). Notons $n = d(x, y)$ et $m = d(y, z)$. Soient $(x, a_1, s_1, \ldots, y)$ un plus court chemin de x à y et $(y, a'_1, \ldots, z)$ un plus court chemin de y à z (on peut forcément trouver de tels chemins puisque G est connexe). La suite $(x, a_1, s_1, \ldots, y, a'_1, \ldots, z)$ est un chemin de x à z, qui est de longueur $n + m$. Un plus court chemin de x à z est donc nécessairement de longueur au plus $n + m$, c'est-à-dire que $d(x, z) \leq n + m = d(x, y) + d(y, z)$.

Exercice 8. Soit G un multigraphe non orienté non nul. Montrer que si G est connexe, alors

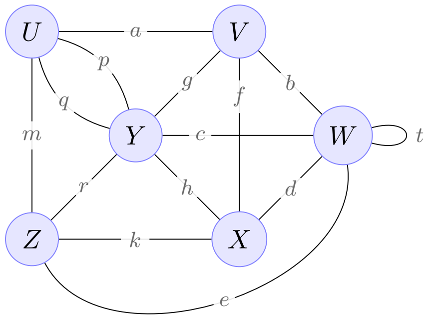
$$\mathrm{ray}(G) \leq \mathrm{diam}(G) \leq 2 \cdot \mathrm{ray}(G)$$

La première inégalité suit directement de la définition, puisque $\mathrm{ray}(G)$ et $\mathrm{diam}(G)$ sont respectivement le minimum et le maximum de l'ensemble $\{\mathrm{exc}(s) \mid s \in S_G\}$.

Pour la deuxième, prenons deux sommets x et y tels que $d(x, y) = \mathrm{diam}(G)$, et un sommet central c. Par l'inégalité triangulaire de l'exercice précédent, $d(x, y) \leq d(x, c) + d(c, y)$. L'excentricité de c étant la distance maximale entre c et tout autre sommet, on a par définition $d(x, c) \leq \mathrm{exc}(c)$ et $d(c, y) \leq \mathrm{exc}(c)$, donc $d(x, y) \leq 2 \cdot \mathrm{exc}(c)$. On conclut en remplaçant $d(x, y)$ par $\mathrm{diam}(G)$ et $\mathrm{exc}(c)$ par $\mathrm{ray}(G)$ (puisque c est central).

Correction TD 6 : Graphes eulériens

Exercice 1. Trouver une tournée eulérienne dans le graphe suivant :



Une possibilité est $(U, a, V, b, W, c, Y, g, V, f, X, d, W, t, W, e, Z, k, X, h, Y, r, Z, m, U, p, Y, q, U)$, ou de manière plus lisible, le chemin qui part de U et dont la suite d'arêtes associée est $(a, b, c, g, f, d, t, e, k, h, r, m, p, q)$.

Exercice 2. Déterminer pour quelles valeurs de $n \in \mathbb{N}$ les graphes des familles suivantes sont eulériens.

1. K_n, les graphes complets d'ordre n.
2. C_n, les graphes-cycles d'ordre n – on considérera que C_1 est le graphe à un sommet et une boucle, et C_2 le graphe à deux sommets et deux arêtes entre eux.
3. W_n, les roues d'ordre n, définies comme suit : W_0 est le graphe nul, et pour tout $n > 0$, on prend C_{n-1} et on ajoute un sommet universel (relié à tous les autres).
4. P_n, les graphes-chemins d'ordre n.
5. les graphes connexes n-réguliers.

1. n impair.
2. dans un graphe-cycle non nul, tous les sommets sont de degré 2 (noter que C_1 et C_2 ne sont pas des graphes simples, mais chaque sommet est quand même de degré 2 – une boucle ajoute 2 au degré de son sommet). C_n est donc eulérien pour tout $n > 0$.
3. à part pour le cas vraiment dégénéré $n = 1$, aucune valeur de n ne fonctionne, car les sommets de C_{n-1} sont tous de degré 3.
4. ça ne marche pas pour $n = 0$ ni pour aucune valeur de $n > 1$: les extrémités sont de degré 1.
5. ça marche quand n est pair et que le graphe n'est pas nul.

Exercice 3. L'algorithme de Hierholzer prend en entrée un graphe non orienté connexe non nul dont tous les sommets sont de degré pair, et renvoie une tournée eulérienne dans ce graphe.

Il fonctionne comme suit :

— choisir un sommet de départ s et construire un chemin fermé simple C

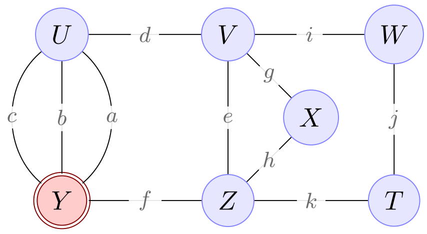
— tant qu'il reste des arêtes dans le graphe qui ne sont pas dans C :

  — choisir un sommet s' dans C qui est incident à une arête non utilisée

  — en partant de s', et en n'utilisant que des arêtes non utilisées, construire un chemin fermé simple C'

— rallonger C en lui insérant C' au niveau du sommet s'

Dérouler l'algorithme de Hierholzer pour construire une tournée eulérienne dans le graphe suivant, en commençant au sommet indiqué.



Exercice 4. On dit qu'un multigraphe non orienté est *semi-eulérien* s'il a un chemin eulérien ouvert. (NB : la définition de « semi-eulérien »varie, dans certains textes l'existence d'un chemin eulérien suffit.)

1. Le graphe de l'exercice 1 est-il semi-eulérien ?
2. Si on lui ajoute une arête s entre U et V, devient-il semi-eulérien ?
3. Et si on ajoute en plus une arête j entre Y et X ?
4. Trouver et prouver une condition nécessaire et suffisante pour qu'un multigraphe non orienté connexe soit semi-eulérien.

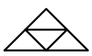
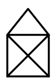
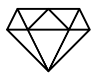
1. Non
2. Oui : on peut reprendre la tournée eulérienne trouvée dans l'exo 1, la faire commencer et finir par U, et ajouter s à la fin.
3. Non, car si on commence en U et qu'on finit en V, on ne peut pas passer une seule fois par toutes les arêtes incidentes à Y.
4. Pour qu'un multigraphe non orienté connexe soit semi-eulérien, il faut et il suffit qu'il ait exactement 2 sommets de degré impair.

   Preuve : soit G un multigraphe non orienté connexe.

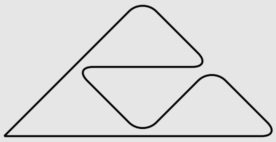
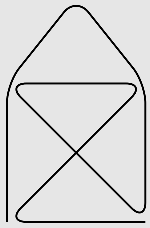
   ($\Rightarrow$) Supposons qu'il existe un chemin eulérien ouvert dans G, et notons ses extrémités s et s'. Si on ajoute une arête $\{s, s'\}$ à G, il devient eulérien, puisqu'il suffit d'ajouter l'arête en question à notre chemin pour avoir une tournée eulérienne. Donc d'après le théorème de caractérisation des graphes eulériens, tous les sommets du graphe modifié sont de degré pair, et donc tous sauf s et s' sont de degré pair dans G.

   ($\Leftarrow$) Supposons que G ait exactement 2 sommets de degré impair ; notons-les s et s'. Si on ajoute une arête $\{s, s'\}$, alors tous les sommets sont de degré pair, et le graphe modifié est eulérien d'après le théorème de caractérisation. On peut donc trouver une tournée eulérienne dans le graphe modifié. Si on la fait commencer à s en plaçant en dernier l'arête ajoutée, lorsqu'on enlève ladite arête, on obtient un chemin de s à s' qui passe une et une seule fois par toutes les arêtes du graphe initial.

Exercice 5. Pouvez-vous dessiner les figures suivantes sans lever le crayon et sans repasser sur un trait déjà dessiné ?

1. 
2. 
3. 

Pour que ce soit possible, il faut un graphe eulérien ou semi-eulérien – autrement dit, il faut qu'il y ait au plus 2 sommets de degré impair.

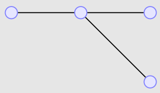
1. 
2. 
3. pas possible, il y a > 2 sommets de degré impair

Exercice 6. Refaire l'exercice 2 pour les graphes semi-eulériens.

1. K_2 est le seul semi-eulérien de la famille.
2. C_n n'est jamais semi-eulérien.
3. W_0 et W_1 ne sont pas semi-eulériens (ils n'ont pas de chemin ouvert). W_2 et W_3 sont semi-eulériens, mais aucun des suivants : en effet, il y a toujours au moins $n-1$ sommets de degré impair dans une roue d'ordre $n > 0$.
4. P_n est toujours semi-eulérien quand $n \geq 2$.
5. Les seuls graphes n-réguliers connexes qui sont semi-eulériens sont les graphes d'ordre 2 quand n est impair.

Exercice 7. 1. Un multigraphe connexe peut-il être à la fois eulérien et semi-eulérien ?

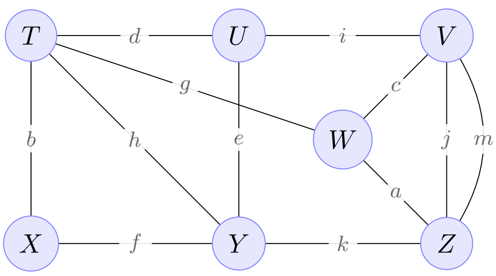
2. Trouver le plus petit multigraphe connexe non nul (d'ordre minimal et de taille minimale, à isomorphisme près) qui ne soit ni eulérien ni semi-eulérien.

1. Non, car pour avoir tous ses sommets de degré pair (condition pour qu'un graphe connexe soit eulérien), il faut n'en avoir aucun de degré impair, ce qui contredit la condition « avoir exactement 2 sommets de degré impair »pour qu'un graphe connexe soit semi-eulérien.
2. Il s'agit du graphe S_4 : 

   Justification (non exigée) : Pour qu'un graphe connexe non nul ne soit ni eulérien ni semi-eulérien, son nombre de sommets de degré impair ne doit être ni 0 ni 2. Mais par le lemme des poignées de main, ce nombre est forcément pair, donc aucun graphe d'ordre $n < 4$ ne respecte la condition (autrement dit, tout graphe connexe d'ordre au plus 3 est soit eulérien, soit semi-eulérien).

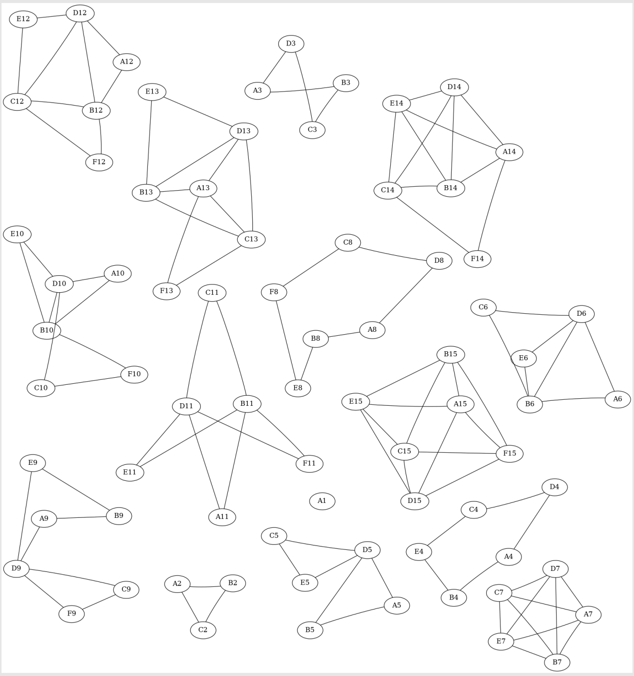
   Essayons de construire un graphe avec 4 sommets. Pour qu'il soit connexe, la solution la plus économique en arêtes est de construire S_4, qui n'a que 3 arêtes. Or S_4 a quatre sommets de degré impair, il n'est donc ni eulérien ni semi-eulérien.

Exercice 8. Proposer une modification de l'algorithme de Hierholzer pour trouver un chemin eulérien dans un graphe semi-eulérien. Déroulez votre algorithme sur le graphe suivant :



Il suffit de procéder comme dans la preuve : on trouve les deux sommets de degré impair, on ajoute une arête entre eux, et on applique l'algo initial en commençant par cette arête. Une fois que c'est fini, il suffit d'enlever l'arête pour obtenir un chemin eulérien.

Exercice 9. Dessiner tous (à isomorphisme près) les graphes connexes simples eulériens d'ordre inférieur ou égal à 6 (indice : il y en a 15).



Exercice 10. Le fait que tous les sommets soient de degré pair ne suffit pas à caractériser les graphes eulériens (la condition n'est suffisante que pour les graphes connexes non nuls).

1. Un graphe non orienté qui n'est pas connexe peut-il être eulérien ?
2. Trouver une condition nécessaire et suffisante pour qu'un graphe non orienté (connexe ou non) soit eulérien.

1. Oui, notamment tout graphe sans arête est trivialement eulérien
2. il doit y avoir au moins un sommet, tous les sommets doivent être de degré 2, et toutes les arêtes doivent appartenir à une même composante connexe.