

Répondez directement sur le sujet. Insérez ensuite votre réponse dans une copie d'examen comportant tous les renseignements administratifs.

Cette épreuve comporte 6 exercices et un aide-mémoire.

Indiquez votre numéro d'anonymat :

Rappel :

- Un graphe *complet* est un graphe sans boucle tel que chaque sommet est relié à chaque autre sommet par exactement une arête.
- Un graphe *simple* est un graphe qui n'a ni boucle, ni arête multiple.

Exercice 1 :

Questions de cours (toute réponse non justifiée ne sera pas prise en compte)

Le nombre maximal d'arêtes d'un graphe simple de n sommets est $n(n-1)$

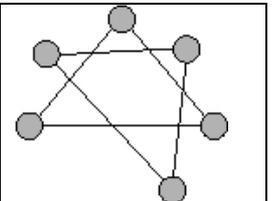
- OUI
 NON

Justifiez :
Le graphe étant simple, le degré maximal de chaque sommet est $n-1$; la somme des degrés est donc, au maximum, égale à $n(n-1)$; ce nombre est égal à 2 fois le nombre d'arêtes. **On a donc au maximum $n(n-1)/2$ arêtes.**

Si un graphe n'a que des sommets pairs alors il admet un cycle eulérien.

- OUI
 NON

Justifiez :
Un graphe peut avoir 0 sommet impair et ne pas être connexe.
Dans ce cas il n'est pas eulérien et n'admet pas de cycle eulérien.
Par exemple le graphe ci-contre.



Un graphe peut avoir 1 seul sommet impair

- OUI
 NON

Justifiez :
C'est impossible car une conséquence du théorème sur la somme des degrés des sommets d'un graphe est que le nombre de sommets impairs est nécessairement pair.

Si n est impair, un graphe complet de n sommets est eulérien.

- OUI
 NON

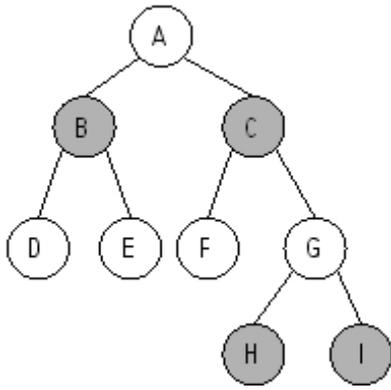
Justifiez :
Dans un graphe complet de n sommets, tous les sommets ont pour degré $n-1$. Donc, si n est impair, tous les sommets sont de degré pair. De plus un graphe complet est connexe. Un tel graphe est donc eulérien.

Si un graphe est sans cycle alors il est 2-coloriable.

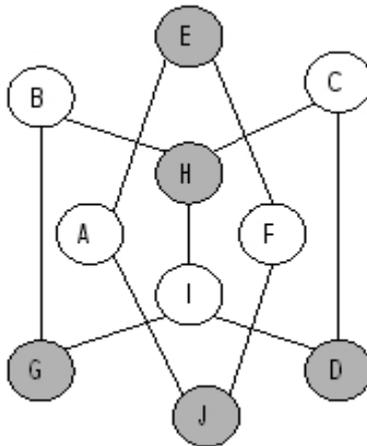
- OUI
 NON

Justifiez :
Si un graphe est sans cycle il n'a donc pas de cycle de longueur impaire. Il est donc 2-coloriable.

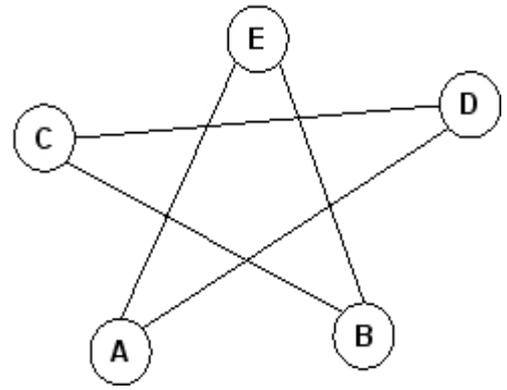
Exercice 2 :



G1



G2



G3

Pour les graphes G1, G2 et G3 cochez les affirmations exactes et justifiez votre réponse. De plus pour les graphes 2-coloriables donnez une coloration sur le dessin et pour les graphes eulériens donner une chaîne ou un cycle eulérien.

	G1	G2	G3
Est connexe	<input checked="" type="checkbox"/> Oui <input type="checkbox"/> Non Pour tout sommet du graphe, il existe une chaîne entre A et ce sommet.	<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non Il n'existe pas de chaîne entre A et B. Il y a deux composantes connexes : BCDGIH et AEFJ.	<input checked="" type="checkbox"/> Oui <input type="checkbox"/> Non Pour tout sommet du graphe, il existe une chaîne entre A et ce sommet.
Est eulérien	<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non Ce graphe a 8 sommets impairs : D, E, F, H, I sont de degré 1 et B, C, G de degré 3	<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non Il n'est pas connexe donc pas eulérien.	<input checked="" type="checkbox"/> Oui <input type="checkbox"/> Non Il est connexe et il a 0 sommet impair. Un exemple de cycle eulérien : AEBCDA*.
Est complet	<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non Il n'existe pas d'arête entre le sommet D et le sommet E.	<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non Il n'existe pas d'arête entre le sommet A et le sommet B.	<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non Il n'existe pas d'arête entre le sommet A et le sommet B.
Est 2-coloriable	<input checked="" type="checkbox"/> Oui <input type="checkbox"/> Non Il n'a pas de cycle impair. Voir coloration ci-dessus.	<input checked="" type="checkbox"/> Oui <input type="checkbox"/> Non Il n'a pas de cycle impair. Voir coloration ci-dessus.	<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non Le cycle AEBCDA* a pour longueur 5. Le graphe a au moins un cycle impair.

* ce graphe étant sans arête multiple une chaîne peut être décrite sans ambiguïté par la liste des sommets.

Aide mémoire fourni pour l'examen : fonctions de manipulation de graphes

Fonction et arguments	Fonctionnalité
<code>nom_graphe(G)</code>	Retourne le nom du graphe G .
<code>nb_sommets(G)</code>	Retourne le nombre de sommets du graphe G .
<code>liste_sommets(G)</code>	Retourne la liste des sommets du graphe G .
<code>chercher_sommet(G,nom)</code>	Retourne le sommet de G qui s'appelle <code>nom</code> , ou <code>None</code> (qui signifie «aucun») s'il n'y a aucun sommet avec ce nom.
<code>nom_sommet(s)</code>	Retourne le nom du sommet s .
<code>degre(s)</code>	Retourne le degré du sommet s .
<code>liste_voisins(s)</code>	Retourne la liste des voisins du sommet s .
<code>marquer_sommet(s)</code>	Marque le sommet s .
<code>demarquer_sommet(s)</code>	Démarque le sommet s .
<code>est_marque(s)</code>	Retourne <code>True</code> si le sommet s est marqué, <code>False</code> sinon.
<code>colorier(s,c)</code>	Colorie le sommet s avec la couleur c . Les noms de couleurs sont des chaînes de caractères, en anglais. Les couleurs disponibles sont les suivantes : "red", "green", "blue", "orange", "cyan", "magenta", "yellow", "white".
<code>couleur(s)</code>	Retourne la couleur du sommet s .
<code>sommet(G,i)</code>	Retourne le sommet numéro i du graphe G , où i est un entier compris entre 0 et <code>nb_sommets(G)-1</code> .
<code>voisin(s,i)</code>	Autour d'un sommet s , les brins d'arêtes sont numérotés de 0 à <code>degre(s)-1</code> . La fonction retourne le sommet obtenu à partir de s en suivant l'arête numéro i , avec cette numérotation.

Exercice 3 :

1. Ecrire une fonction

`somme_diviseurs_stricts(n)` :
qui pour un entier n ($n > 1$) retourne la
somme de ses diviseurs stricts (on
appelle diviseurs stricts de n ceux qui
sont strictement plus petits que n)

```
def somme_diviseurs_stricts(n) :  
    s = 0  
    for i in range(1, n) :  
        if n%i == 0:  
            s = s + i  
    return s
```

Un entier n est dit « *parfait* » s'il
est égal à la somme de ses
diviseurs stricts. Par exemple, 6 et
28 sont des nombres parfaits.

Ecrire une fonction

`entier_parfait(n)` qui
renvoie `True` si l'entier n est
parfait, et `False` sinon.

```
def entier_parfait(n) :  
    return somme_diviseurs_stricts(n) == n
```

Exercice 4 :

1. Pour un graphe G , rappeler la relation qui existe entre les degrés des sommets du graphe et le nombre de ses arêtes.

La somme des degrés des sommets d'un graphe est égale à deux fois le nombre d'arêtes.

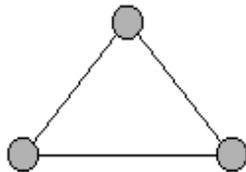
$$\sum_{s \in S(G)} d(s) = 2 \times |A(G)|$$

2. En déduire quelle condition doit être vérifiée afin que le nombre de sommets de G soit égal au nombre d'arêtes de G ?

La somme des degrés des sommets de ce type de graphe doit être égale à deux fois le nombre de sommets.

3. Dessiner un exemple de graphe dont le nombre de sommet est égal au nombre d'arêtes.

3 sommets, 3 arêtes.



4. Ecrire une fonction `meme_nombre_arettes_sommets(G)` qui renvoie `True` si le graphe G a autant de sommets que d'arêtes, et `False` sinon.

```
def meme_nombre_arettes_sommets(G) :  
    somme = 0  
    for s in liste_sommets(G) :  
        somme = somme + degre(s)  
    nb_arettes = somme/2  
    return nb_sommets(G) == nb_arettes
```

Exercice 5 :

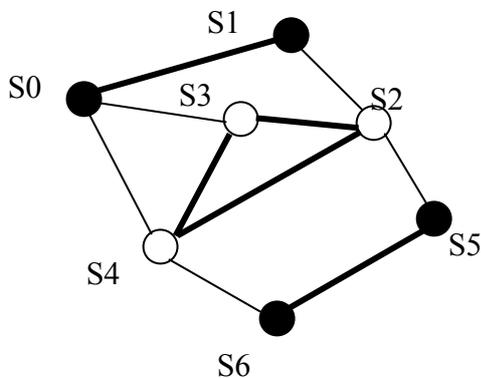
1. Ecrire une fonction `nb_voisins_meme_couleur(s)` qui retourne le nombre de voisins de `s` qui sont de la même couleur que `s`.

```
def nb_voisins_meme_couleur(s) :  
    nb = 0  
    for v in liste_voisins(s) :  
        if couleur(v) == couleur(s):  
            nb = nb + 1  
    return nb
```

On considère la fonction `mystere(G)` suivante :

```
def mystere(G) :  
    cpt=0  
    for s in liste_sommets(G) :  
        cpt=cpt+nb_voisins_meme_couleur(s)  
    return cpt/2
```

2. Faire tourner la fonction `mystere` sur le graphe `G4` ci-dessous dont les sommets sont coloriés soit en noir soit en blanc.



S	nb sommets meme couleur(s)	cpt
		0
S0	1	1
S1	1	2
S2	2	4
S3	2	6
S4	2	8
S5	1	9
S6	1	10

3. Quelle valeur retourne `mystere(G4)` ?

5

4. Pour un graphe `G`, que retourne `mystere(G)` ?

La fonction `mystere` retourne le nombre d'arêtes dont les extrémités sont de la même couleur. Voir dessin ci-dessus.

Exercice 6 :

Sept pays sont représentés ci-dessous avec leurs frontières :

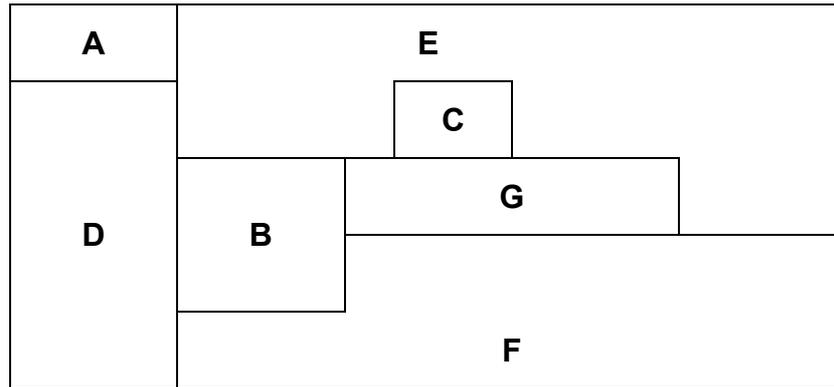
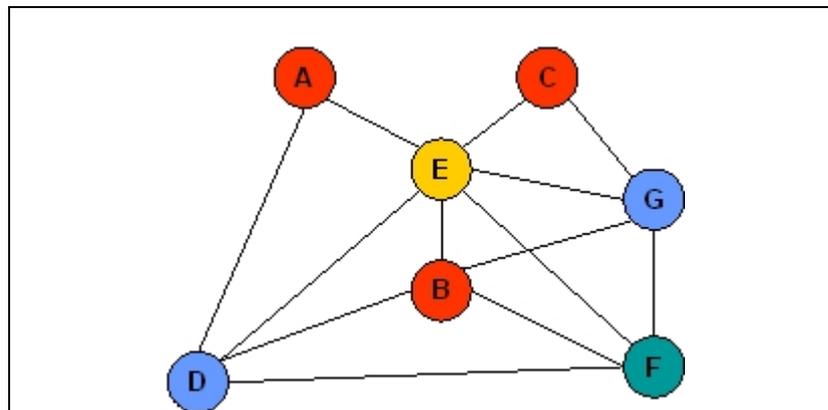


figure 1

1- Modéliser à l'aide d'un graphe la relation "a une frontière avec".



2- Quel est le nombre minimum de couleurs nécessaires pour colorier cette carte de sorte que deux pays voisins ne soient pas coloriés de la même couleur ? Justifier dans la case ci-dessous et colorier le graphe ci-dessus associé à la carte de la figure 1.

Deux pays voisins correspondent à deux sommets voisins dans le graphe G ci-dessus.
Une coloration correcte de la carte correspond à une coloration du graphe G.
Le graphe G contient un sous-graphe complet à 4 sommets : E, B, F, G. Il faudra donc au minimum 4 couleurs pour colorier ce graphe.
La coloration ci-dessus montre qu'il est 4-coloriable.

3- Est-il possible de visiter tous les pays en traversant chaque frontière une et une seule fois ? Justifier

Dans le graphe G, une arête entre deux sommets correspond à l'existence d'une frontière commune entre les deux pays représentés par les sommets. Donc chercher un itinéraire permettant de visiter tous les pays au moins une fois en traversant chaque frontière une et une seule fois revient à chercher une chaîne eulérienne dans le graphe G.
Une telle chaîne existe car le graphe G est eulérien, en effet il est connexe et il n'a pas de sommet impair.

4- Si oui, toujours en visitant tous les pays et en traversant chaque frontière une seule fois, peut-on revenir à son point de départ ? Justifier.

Le graphe G étant eulérien avec 0 sommet impair, toutes les chaînes eulériennes sont des cycles.
Par exemple : EADEBDFBGFE (Le graphe G étant sans arête multiple une chaîne peut être décrite sans ambiguïté par la liste des sommets)
Pour tout itinéraire permettant de visiter au moins une fois chaque pays en traversant chaque frontière une seule fois, on revient toujours à son point de départ.