



ANNEE UNIVERSITAIRE 2008/2009
1^{ère} SESSION D'AUTOMNE

Parcours : MISMI UE:MSI102 Epreuve : Informatique
Date : 6 janvier 2009 -11h-12h30 Durée : 1h30
Documents : Aucun document autorisé
Epreuve de : C. Blanc, G. Melançon, C. Pannier, M. Zeitoun



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 :

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

OUI

NON

Justifiez :

Un graphe peut avoir 1 seul sommet impair

OUI

NON

Justifiez :

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

OUI

NON

Justifiez :

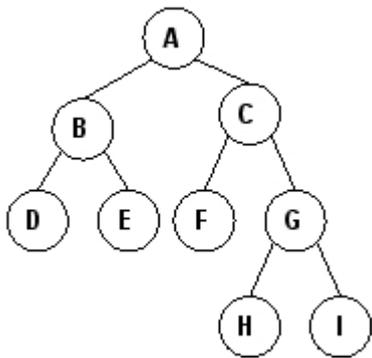
Si un graphe est sans cycle alors il est 2-coloriable.

OUI

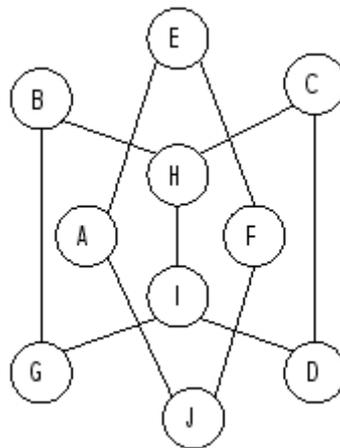
NON

Justifiez :

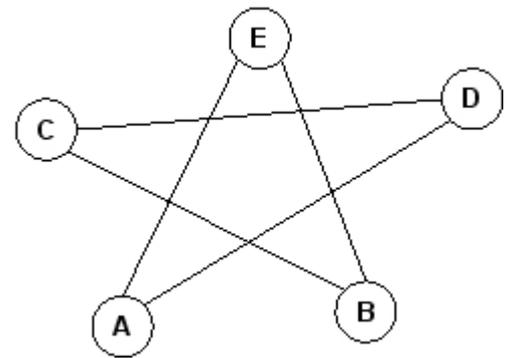
Exercice 2 :



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 type="checkbox"/> Oui <input type="checkbox"/> Non	<input type="checkbox"/> Oui <input type="checkbox"/> Non	<input type="checkbox"/> Oui <input type="checkbox"/> Non
Est eulérien	<input type="checkbox"/> Oui <input type="checkbox"/> Non	<input type="checkbox"/> Oui <input type="checkbox"/> Non	<input type="checkbox"/> Oui <input type="checkbox"/> Non
Est complet	<input type="checkbox"/> Oui <input type="checkbox"/> Non	<input type="checkbox"/> Oui <input type="checkbox"/> Non	<input type="checkbox"/> Oui <input type="checkbox"/> Non
Est 2-coloriable	<input type="checkbox"/> Oui <input type="checkbox"/> Non	<input type="checkbox"/> Oui <input type="checkbox"/> Non	<input type="checkbox"/> Oui <input type="checkbox"/> Non

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 nom, ou None (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 True si le sommet s est marqué, False 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)

2. 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.

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.

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

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

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

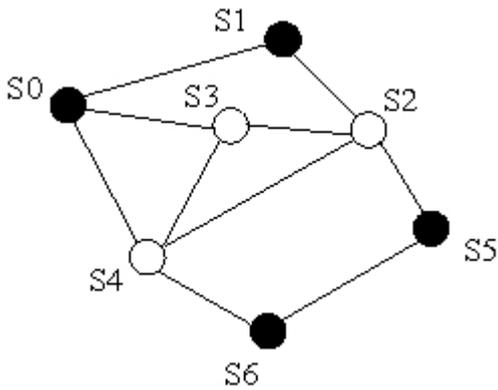
Exercice 5 :

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`.

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.



G4

S	nb_sommets_meme_couleur(s)	cpt
		0
S0		
S1		
S2		
S3		
S4		
S5		
S6		

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

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

Exercice 6 :

Sept pays sont représentés ci-dessous avec leurs frontières :

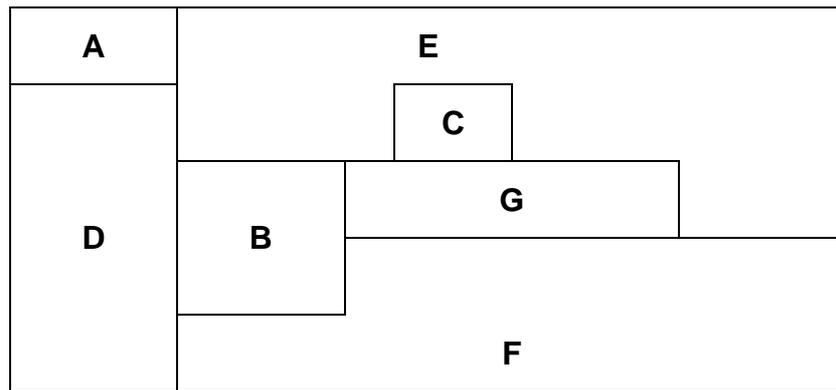
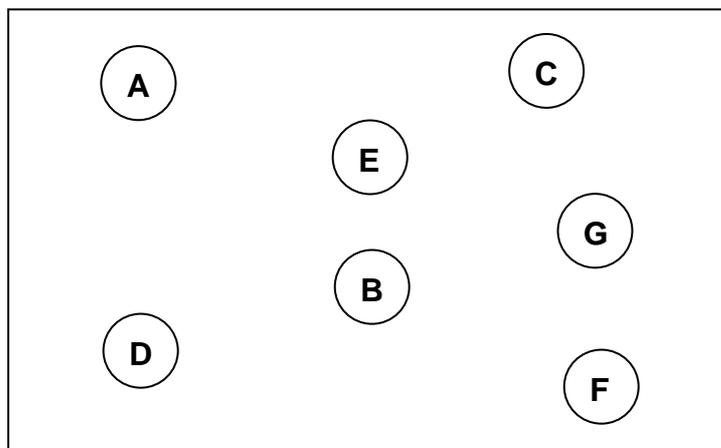


figure 1

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.

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

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.