Liens
Supports pédagogiques cours et exercices
https://drive.google.com/drive/u/0/folders/1IHsJA4WINKqjkiW3v4udx4EUYR8c6Uzv

composition & répertoire de travail de votre équipe : email de M. Lapoire du 24/1/2022

Séance 2

LECTURE
Lire attentivement le chapitre 1 du cours.

EXERCICES A RENDRE : page suivante

Rappels : pour tout entier n, [1,n] désigne l'intervalle d'entiers {1,2,..,n} ; [1,0] désigne ∅. Une *permutation* est une bijection d'un ensemble vers lui-même. Une *paire* est un ensemble de cardinalité 2. Ainsi, {1,2} est une paire ; {1,1} n'en est pas une!. Deux ensembles sont *disjoints* si leur intersection est vide.

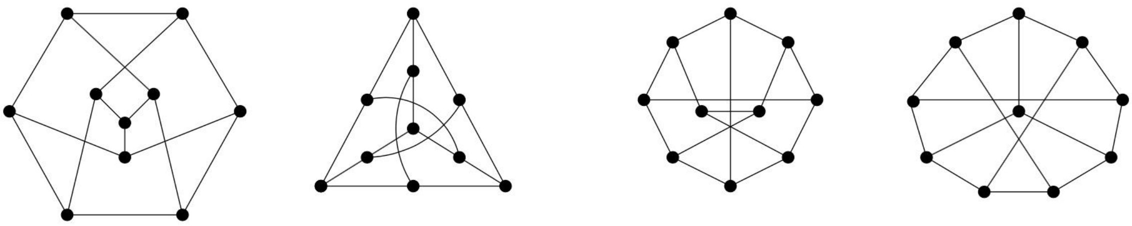
Considérons le problème suivant :

```
AdmetDessinSansNom
Entrée :   G un graphe
           D un dessin de graphes sans noms sur les sommets

Sortie : le booléen (G admet pour dessin D augmenté de noms de
sommets)
```

exercice 1

Considérons D1, D2, D3 et D4 les 4 dessins sans noms sur les sommets :



Est-ce que pour chaque Di, le graphe mystère G de la séance 1 admet Di comme dessin? Fournir la preuve. Ce qui revient à calculer les 4 booléens :
`AdmetDessinSansNom(G,Di)`

exercice 2

Nous admettrons ici que les algorithmes suivants sont de complexités en temps linéaires :

- `fonction graphe2gessin(G:graphe):dessin` qui retourne un dessin du graphe G.
- `fonction dessin2graphe(D:dessin):graphe` qui retourne le graphe de dessin (complet) D.
- `fonction effaceNoms(D:dessin):dessinSansNom` qui efface les noms des sommets sur un dessin d'un graphe.

Ainsi que d'autres algorithmes que vous définirez précisément.

1. Démontrer que si le problème `admetDessinSansNom` admet une solution en temps polynomial alors `SONT_ISOMORPHES` aussi.
   Indication : écrire une solution algorithmique de `SONT_ISOMORPHES` utilisant une fonction résolvant `admetDessinSansNom`.
2. Que pensez-vous de la complexité en temps d'une solution de `admetDessinSansNom` ?
3. Démontrer que si le problème `SONT_ISOMORPHES` admettait une solution en temps polynomial alors `AdmetDessinSansNom` aussi.
   Indication : écrire une solution algorithmique de `admetDessinSansNom` utilisant une fonction résolvant `SONT_ISOMORPHES`.

4. Que pensez-vous de la complexité en temps d'une solution de `admetDessinSansNom` ?

Pour l’exercice suivant, nous supposons disposer d’une fonction de complexité en temps linéaire qui résout le problème suivant :

```
est3coloration
Entrée :   une fonction f VG→[1,3], G un graphe
Sortie : le booléen (f est une 3-coloration de G)
```

exercice 3 (8 minutes)

Nous supposons ici que tout ensemble de sommets d’un graphe est un intervalle d’entier de la forme [1,n] avec n entier.

1. Comment représenter sur machine très simplement toute fonction f : $V_G \to [1,3]$ de façon à écrire très simplement l’algorithme `est3coloration ?`
2. Soit G un graphe d’ensemble de sommets V_G = [1,n], combien existe t-il de fonctions f : $V_G \to [1,3]$?
3. Comment représenter sur machine très simplement toute fonction f : $V_G \to [1,3]$ de façon à les calculer très simplement (c.a.d selon un code d’écriture très très simple).
4. Ecrire une solution algorithmique d’écriture (très très) simple au problème `est3colorable` utilisant comme fonction auxiliaire `est3coloration`.
5. Evaluer la complexité en temps de votre algorithme.

exercice 4 (10 minutes)

Nous supposons ici que tout sommet d’un hypercube H_N est représenté par un entier compris entre 0 et 2^N-1 (qui admet pour représentation binaire le mot de N bits).

1. Dessiner les hypercubes H0, H1, H2 et H4.
   Considérons l’hypercube H_{10}
2. Quelle est la distance entre les sommets 0101000111 et 0110100010 ?
3. Fournir un plus court chemin allant de 0101000111 à 0110100010.
4. Fournir l’algorithme calculant la distance entre deux sommets de H_N

exercice 5 (x minutes)

L’objectif de cet exercice est de fournir une preuve (correcte et complète!) de la propriété 5 caractérisations d’un arbre.

Pour établir les équivalences : 1.⇔2.⇔3.⇔>4.⇔5.⇔6. , il suffit par exemple d’établir les 6 implications 1.⇒2.⇒3.⇒>4.⇒5.⇒6.⇒1. Pour cela, nous allons répartir au sein du groupe ce travail de la façon suivante :

1. Chaque équipe établit une preuve directe de 1.⇒2.
2. Chaque équipe établit l’implication déterminée ainsi par le numéro d’équipe :
   a. équipe 1 : implication 2.⇒3.
   b. équipe 2 et 3 : implication 3.⇒4.
   c. équipe 4 et 5 : implication 4.⇒5.
   d. équipe 6 et 8 : implication 5.⇒6.
   e. équipe 7 : implication 6.⇒1.

Indication : contrairement à l’implication 1.⇒2. qui peut être réalisée directement, plusieurs des implications nécessitent une preuve par récurrence.