Faculté des Sciences
Aix*Marseille Université

Année universitaire 2021/2022

Site : ☒ Luminy ☐ St-Charles ☐ St-Jérôme ☐ Cht-Gombert ☐ Aix-Montperrin ☐ Aubagne-SATIS
Sujet de : ☒ 1er semestre ☐ 2ème semestre ☐ Session 2 Durée de l'épreuve : 3h
Examen de : M1 Nom du diplôme : Master Informatique
Code du module : SINAU03L Libellé du module : Complexité
Calculatrices autorisées : NON Documents autorisés : NON

Préambule : Il a été constaté lors de la correction du partiel que certaines copies étaient parfois très difficiles à déchiffrer. Aussi, il vous est demandé de veillez à rendre votre copie aussi claire et lisible que possible. Par ailleurs, il est également arrivé que des algorithmes soient présentés en utilisant parfois des spécificités propres à certains langages de programmation. Aussi, il vous est demandé ici de vous limiter à l'emploi d'un langage algorithmique de base, tel que celui présenté dans les corrigés mis en ligne sur Ametice (pour information, certaines opérations propres à certains langages de programmation peuvent cacher parfois des coûts de traitement mal interprétés lors des analyses de la complexité).

1 Quelques questions de cours

Chaque question peut posséder 0 ou plusieurs réponses possibles. Il vous est juste demandé de préciser "*OUI*" ou "*NON*", ou encore "*Je ne sais pas*", avec les numéros des différents items, ceci sans donner de justification. Mais attention, comme dans certains QCM, la notation prendra en compte (négativement) les réponses erronées. Vous êtes donc invités à ne donner que les réponses dont vous êtes sûrs.

Question 1. Supposons que 2-SAT se réduise avec une réduction many-one polynomiale à SAT. Que peut-on déduire ?

1. $\mathbf{P} = \mathbf{NP}$
2. $\mathbf{P} \neq \mathbf{NP}$
3. $\mathbf{P} = \mathbf{co\text{-}NP}$
4. $\mathbf{P} \neq \mathbf{co\text{-}NP}$

Question 2. Soit A un langage appartenant à la classe **P**. Que peut-on déduire ?

1. Il existe une machine de Turing non déterministe qui reconnaît le langage A en temps polynomial
2. Il existe une machine de Turing déterministe qui reconnaît le langage A en temps exponentiel
3. A est **NP**-complet
4. S'il existe une machine de Turing non déterministe qui reconnaît le langage A en temps polynomial, alors $\mathbf{P} = \mathbf{NP}$

Question 3. Soit A, B et C trois langage appartenant à la classe **NP**. Supposons que A se réduit à B et que B se réduit à C avec deux réductions many-one polynomiales. Que peut-on déduire ?

1. si A est **NP**-complet alors C est **NP**-complet
2. si A est **NP**-complet alors B est **NP**-complet
3. si C est **NP**-complet alors B est **NP**-complet
4. si C est **NP**-complet alors A est **NP**-complet

2 Une autre question de cours

Question. Montrez que si $\mathbf{NP} \neq \mathbf{coNP}$, alors $\mathbf{P} \neq \mathbf{NP}$. Si vous utilisez des résultats connus par ailleurs, veillez à les rappeler précisément.

3 Appartenance à la classe NP

Nous considérons ici le problème appelé "Deux-Ou-Un-3-SAT", que nous noterons "DOU-3-SAT". Dans ce problème, on dispose d'une formule 3-SAT classique et il s'agit de savoir si cette formule possède un modèle pour lequel dans chaque clause, il existe au moins un littéral vrai et un littéral faux (on peut donc avoir pour un tel modèle, soit un littéral vrai et deux littéraux faux, soit deux littéraux vrai et un littéral faux) :

DOU-3-SAT
Donnée : Une formule F de la logique des propositions exprimée sous forme normale conjonctive (FNC) par une FNC dont toutes les clauses possèdent 3 littéraux exactement.
Question : La formule F possède un modèle tel que pour chaque clause, il existe au moins un littéral vrai et un littéral faux ?

Comme représentation des formules, utilisez le format suivant : une formule de m clauses sur les variables $x_1, \ldots, x_n$ est représentée par une matrice de taille $m \times 3$, où chaque ligne contient les trois littéraux de la clause : `F.C[i][j] = k` si le j-ème littéral de la i-ème clause est x_k, et `F.C[i][j] = -k` si le j-ème littéral de la i-ème clause est $\neg x_k$. Le nombre de clauses et le nombre de variables sont dénotes, respectivement, par `F.n` et `F.m`.

Question 1. Montrez que le problème **DOU-3-SAT** appartient à **NP** en utilisant un algorithme non-déterministe.

Question 2. Montrez que le problème **DOU-3-SAT** appartient à **NP** en exploitant la notion de certificat polynomial.

4 Réduction

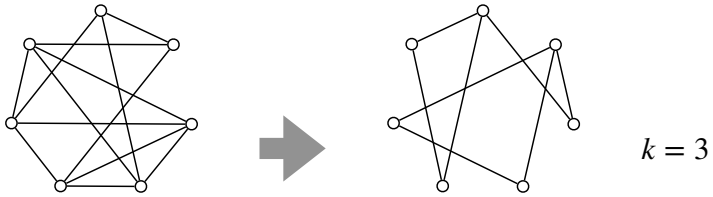
Considérons les deux problèmes suivants :

3-Coloration
Donnée : Un graphe non orienté $G = (S, A)$ sans boucles.
Question : G admet-il une 3-coloration, c'est-à-dire, une façon de colorier les sommets de G avec trois couleurs de telle façon que deux sommets reliés par une arête soient toujours de couleurs différentes ?

Partition en cliques
Donnée : Un graphe non orienté $G = (S, A)$ sans boucles et un entier naturel k.
Question : Existe-t-il une partition en k sous-ensembles $S_1, \ldots, S_k$ de S (c'est-à-dire, l'union de $S_1, \ldots, S_k$ donne S et chaque S_i est disjoint de S_j pour tout $i \neq j$) telle que chaque S_i est une clique (c'est-à-dire, chaque sommet de S_i est relié par une arête à tout autre sommet de S_i) ?

Pour ces problèmes, utilisez la représentation des graphes par matrices d'adjacence.

On veut réduire le problème **3-Coloration** à **Partition en cliques** avec une réduction *many-one* en temps polynomial. Nous donnons ci-dessous un exemple de transformation pour le graphe G_1 suivant, qui se transforme en une instance (G_2, k) du problème **Partition en cliques** avec $k = 3$:



G_1 G_2

Question 1. Trouvez une 3-coloration des sommets de G_1 et une partition de G_2 en trois cliques qui correspond à cette 3-coloration.

Question 2. Trouvez une partition de G_2 en trois cliques, différente de celle de la question précédente, et une 3-coloration de G_1 qui correspond à cette partition.

Question 3. Trouvez un graphe G_3 de 6 sommets qui n'admet pas de 3-coloration (en le justifiant). Quel est le graphe correspondant selon la transformation utilisée précédemment pour G_1 ?

Question 4. Généralisez la transformation du graphe précédent G_1 en paire (*graphe*, *entier*) à des graphes quelconques, en fournissant un algorithme en pseudo-code qui fonctionne en temps polynomial. Fournissez une analyse de la complexité de cet algorithme.

Question 5. Expliquez pourquoi si le graphe G en entrée admet une 3-coloration, alors la sortie (G', k) consiste en un graphe qui admet une partition en k cliques.

Question 6. Expliquez pourquoi si la sortie (G', k) consiste en un graphe qui admet une partition en k cliques, alors le graphe G en entrée admet une 3-coloration.

Question 7. À partir des réponses aux questions précédentes, que peut-on conclure concernant la transformation proposée ? Justifiez votre réponse.

Question 8. En sachant que **3-Coloration** est **NP**-difficile et qu'on peut résoudre **Partition en cliques** en temps polynomial avec un algorithme non-déterministe, comment peut-on classer **Partition en cliques** ? Justifiez votre réponse.

Question 9. Réduisez le problème **Partition en cliques** au problème **k-Coloration** suivant avec une réduction *many-one* polynomiale, en justifiant chaque passage (pour cette question il n'y a pas besoin d'écrire du pseudo-code détaillé, mais il faut quand même décrire et analyser la réduction de façon non ambiguë) :

> **k-Coloration**
> **Donnée :** Un graphe non orienté $G = (S, A)$ sans boucles et un entier naturel k.
> **Question :** Peut-on colorier les sommets de G avec k couleurs de telle façon que deux sommets reliés par une arête soient toujours de couleurs différentes ?

Question 10. À partir des résultats obtenus avec les questions précédentes, est-il possible de déduire l'existence d'une réduction *many-one* polynomiale de **3-Coloration** vers **k-Coloration** ? Pourquoi ?

Question 11. À partir des résultats obtenus avec les questions précédentes, est-il possible déduire l'existence d'une réduction *many-one* polynomiale de **k-Coloration** vers **3-Coloration** ? Pourquoi ?

Question 12. L'algorithme suivant vérifie si un graphe admet une **2-Coloration** (disons en noir et rouge) :

> On commence par colorier un sommet arbitraire en rouge pour chaque composante connexe du graphe ; tant qu'il reste des sommets s coloriés (en noir ou en rouge) avec des voisins non encore coloriés, coloriez tous ses voisins avec l'autre couleur. Si à un moment donné on découvre une incohérence (il faut colorier en rouge le voisin d'un sommet alors qu'il est déjà en noir, ou vice-versa), alors on rejette le graphe en entrée. Si, au contraire, on arrive à colorier la totalité du graphe sans trouver d'incohérences, alors on accepte.

Expliquez pourquoi, sous l'hypothèse que $\mathbf{P} \neq \mathbf{NP}$, il n'existe pas de réduction *many-one* polynomiale de **Partition en cliques** vers le problème **2-Coloration**.

Question 13. Est-ce qu'il existe une réduction *many-one* polynomiale de **2-Coloration** vers **Partition en cliques** ? Pourquoi ?