Examen (durée : 2h30. Tout document autorisé)

IMPORTANT : veuillez soigner votre rédaction et vos explications. Veuillez bien numéroter les exerices traités et quand cela est nécessaire les illustrer avec des exemples. Il sera tenu compte de la clarté des copies.

Exercice 1 :

1. Classer les fonctions de n suivantes par ordre croissant de complexité :

$$n^2, \log n, (\log n)^2, n^{3/2}, n \log n, \sqrt{n}, 15 \log n, \log \log n\,.$$

2. On a deux fonctions f et g telles que $f(n) = \Omega(\sqrt{n})$ et $g(n) = \mathcal{O}(n \log n)$. Pour chacun des points a), b), c), d) et e), exhibez quand cela s'avère possible (le cas échéant expliciter pourquoi c'est impossible) deux fonctions $f(n)$ et $g(n)$
- qui vérifient les égalités (exemples),
- que ne les vérifient pas (contre-exemples).

a) $f(n) = \mathcal{O}(g(n))$
b) $f(n) = \Omega(\log(g(n)))$
c) $f(n) = \Theta(g(n))$
d) $f(n) + g(n) = \Omega(n^2)$
e) $e^{g(n)} = \Theta(e^{f(n)})$

Exercice 2 :

Dans cet exercice, on note par $C(n)$ le temps nécessaire pour un algorithme de tri pour trier un tableau de taille n. On suppose que C vérifie :

$$C(1) = 0 \text{ et } C(n) \leq c.n + 2C(n/2) \ \ (\text{pour } n \geq 1)\,.$$

1. Montrer que $C(n) = \mathcal{O}(n \log n)$.

Exercice 3 :

Dans cet exercice, étant donné n points du plan (x_1, y_1), (x_2, y_2), $\cdots$, (x_n, y_n) de $\mathbb{R}^2$, on veut calculer la d distance entre les deux points les plus proches.

1. Ecrire un algorithme calculant d.

2. Donner sa complexité.

Exercice 4 :

Dans un groupe de personnes, il semble qu'il y ait toujours deux individus qui connaissent exactement le même nombre de membres du groupe.

1. Formaliser la propriété à démontrer dans le vocabulaire des graphes.

2. Démontrer cette propriété (on pourra notamment raisonner par l'absurde).

Exercice 5 :

Dans cet exercice, on considère le problème de décision suivant :

INPUT : Un graphe $G = (V,\ E)$ et un entier k.

OUTPUT : OUI s'il existe un sous-ensemble $V' \subseteq V$ tel que $|V'| = k$ et pour tout $u, v \in V'$, $(u, v) \notin E$. NON dans le cas contraire.

On appelera ce problème, le problème STABLE. Concrètement, le problème STABLE consiste à décider si OUI ou NON, un graphe G contient un ensemble indépendant de taille k.

a) Donner un exemple de graphe $G = (V, E)$ avec $|V| = n = 5$ sommets et $|E| = 6$ arêtes contenant un ensemble indépendant de taille $k = 3$.

b) Est-il possible d'avoir un graphe $G = (V, E)$ avec $|V| = n = 5$ sommets et $|E| = 6$ arêtes contenant un ensemble indépendant de taille $k = 4$? Justifier votre réponse.

On veut maintenant montrer que STABLE est NP-complet par réduction à 3-SAT.

c) Rappeler le problème de décision 3-SAT en précisant l'INPUT.

d) Montrer que STABLE est NP-complet en utilisant le *gadget* suivant :

A la formule 3-SAT $F = \wedge_{1 \leq i \leq k} c_i$ avec $c_i = x_{1i} \vee x_{2i} \vee x_{3i}$ on associe un graphe G avec $3k$ sommets, un pour chaque occurrence d'un littéral dans une clause :

- pour chaque variable x_i de 3-SAT, G pposs-de une arête entre chaque sommet associé à un littéral x_i et chaque sommet associé à un littéral $\neg x_i$ (un stable de G correspond alors à une affectation de valeurs de vérité à une partie des variables),
- Pour chaque clause c de 3-SAT on associe un triangle (par exemple : si $c = (x_1 \vee \neg x_2 \vee x_3)$ alors G a le triangle $(x_1, \neg x_2)$, $(\neg x_2, x_3)$, (x_3, x_1)).
-

Il est à noter qu'ici il n'y a pas de modélisation à faire mais par contre il faut montrer pourquoi un certificat se vérifie en temps polynomial ("STEP 1") et pourquoi le gadget décrit ci-dessus permet-il de relier la complexité de 3-SAT à celle de STABLE ("STEP 2").