COMPLEX - Partie 2
2021-2022

Énoncé

Nom ou numéro d'anonymat :

Durée : 1 heure 30 minutes

Notes manuscrites et documents de cours autorisés
L'utilisation de tout matériel électronique (en dehors d'une montre non connectée) est interdite

Les exercices sont indépendants.
Une rédaction claire et concise sera appréciée. Toute affirmation devra être justifiée.
Une question non résolue n'empêche pas de faire les suivantes
(dans ce cas indiquez clairement que vous admettez le(s) résultat(s) de la question non faite).

Exercice 1 : Ensemble indépendant

Soit $G = (V, E)$ un graphe non-orienté et sans boucle. Un ensemble indépendant dans G est un sous-ensemble $S \subseteq V$ tels que aucune paire de sommets de S ne sont reliés par une arête de E. Le problème de trouver un ensemble indépendant de taille maximale dans un graphe est un problème NP-complet.

Nous notons $N(v)$ le voisinage de v dans G, c'est-à-dire l'ensemble des sommets u pour lesquels le couple $\{u, v\}$ est une arête de E : $N(v) = \{u \in V | \{u, v\} \in E\}$. Nous notons également $\deg(v) = \#N(v)$ le degré d'un sommet $v \in V$, c'est-à-dire le nombre d'arêtes contenant ce sommet et $\Delta = \max_{v \in V} \deg(v)$, le degré maximal d'un sommet de G.

1.a] Considérons l'algorithme déterministe suivant pour calculer un ensemble indépendant S.

1. initialiser l'ensemble S à $\emptyset$ et l'ensemble T à V
2. tant que T est non vide
   (a) retirer un sommet v de T (dans un ordre arbitraire)
   (b) ajouter v à S
   (c) supprimer les éléments de $N(v)$ de T
3. retourner S

Montrer que cet algorithme retourne bien un ensemble indépendant de G.

1.b] Montrer que l'ensemble S retourné par cet algorithme vérifie $\#S \geq n/(\Delta + 1)$.

1.c] Considérons le graphe $G = (V, E)$ défini par $V = \{1, \ldots, n\}$ et $E = \{\{1, i\}, i \in \{2, \ldots, n\}\}$.
Montrer que G possède un ensemble indépendant à $n - 1$ sommets et que le degré maximal de G est égal à $n - 1$.

Montrer que l'algorithme précédent peut effectivement retourner pour ce graphe un ensemble indépendent S à un seul sommet.

1.d] L'exemple précédent montre que les degrés des sommets d'un graphe peut varier fortement. Il est donc préférable de regarder les degrés des sommets individuellement. Dans la suite nous considérons que l'ensemble des sommets V du graphe $G = (V, E)$ est l'ensemble des entiers consécutifs $\{1, \ldots, n\}$.

Considérons σ une permutation quelconque de l'ensemble $V = \{1, \ldots, n\}$ et notons

$$S_\sigma = \{i \in V \mid \forall j \in N(i), \sigma(i) < \sigma(j)\}$$

Montrer que S_σ est un ensemble indépendant de G.

1.e] Considérons l'algorithme probabiliste qui consiste à tirer uniformément aléatoirement une permutation σ de l'ensemble $V = \{1, \ldots, n\}$ et à retourner l'ensemble S_σ associé. Considérons pour tout sommet i de V, la variable aléatoire

$$X_i = \begin{cases} 1 & \text{si } i \in S_\sigma \\ 0 & \text{sinon} \end{cases}$$

Montrer que $\mathbb{E}(X_i) = 1/(\deg(i) + 1)$.

1.f] Notons X la variable aléatoire correspondant au nombre de sommets de S_σ retourné par l'algorithme probabiliste. Déduire de la question précédente que $\mathbb{E}(X) = \sum_{i=1}^{n} 1/(\deg(i)+1)$.

1.g] En déduire la taille moyenne de l'ensemble indépendant retourné par l'algorithme probabiliste sur le graphe de la question **1.c**.

Exercice 2 : $\mathsf{NP} \subseteq \mathsf{BPP} \Rightarrow \mathsf{NP} = \mathsf{RP}$

2.a] Montrer que $\mathsf{RP} \subseteq \mathsf{NP}$

Le but de l'exercice est de montrer que si $\mathsf{NP} \subseteq \mathsf{BPP}$ alors $\mathsf{NP} \subseteq \mathsf{RP}$ (et donc $\mathsf{NP} = \mathsf{RP}$).

2.b] Rappelons que, dans le cadre des langages formels pour les problèmes de décision sur un alphabet Σ, on dit qu'un langage $\mathcal{L}_1 \subset \Sigma^*$ est *réductible en temps polynomial* à un langage $\mathcal{L}_2 \subset \Sigma^*$ (ce qui est généralement noté $\mathcal{L}_1 \leq_P \mathcal{L}_2$) s'il existe une fonction calculable en temps polynomial $f : \Sigma^* \to \Sigma^*$ telle que pour tout $w \in \Sigma^*$, $x \in \mathcal{L}_1$ si et seulement si $f(x) \in \mathcal{L}_2$.

Soient $\mathcal{L}_1$ et $\mathcal{L}_2$ deux langages sur un alphabet Σ tels que $\mathcal{L}_1 \leq_P \mathcal{L}_2$. Montrer que si $\mathcal{L}_2 \in \mathsf{RP}$ alors $\mathcal{L}_1 \in \mathsf{RP}$.

2.c] Nous supposons que $\mathsf{SAT} \in \mathsf{BPP}$. Plus précisément, avec les techniques d'amplification vues en cours et en TD, nous supposons qu'il existe une machine de Turing probabiliste $\mathcal{M}$ qui prenant en entrée une formule booléenne Φ en n variables $(x_1, \ldots, x_n)$ sous forme normale conjonctive de m clauses retourne un bit de sorte que :

— si Φ est satisfiable, $\Pr[\mathcal{M}(\Phi) = 1] \geq 1 - 4^{-n}$;

— si Φ n'est pas satisfiable, $\Pr[\mathcal{M}(\Phi) = 0] \geq 1 - 4^{-n}$.

Considérons la machine de Turing probabiliste $\mathcal{N}$ qui exécute l'algorithme suivant :

1. initialiser une formule booléenne Ψ à Φ
2. pour i de 1 à $n - 1$
   (a) construire la formule booléenne $\Psi_{x_i=0}$ obtenue en remplaçant chaque clause par la clause obtenue en fixant la valeur x_i à 0 (c'est-à-dire les clauses où x_i apparaît sous la forme d'un littéral positif $x_i \vee \ell_1 \vee \cdots \vee \ell_t$ sont remplacées par $\ell_1 \vee \cdots \vee \ell_t$ et les clauses où x_i apparaît sous la forme d'un littéral négatif $\neg x_i \vee \ell_1 \vee \cdots \vee \ell_t$ sont supprimées).
   (b) exécuter la machine de Turing $\mathcal{M}$ sur $\Psi_{x_i=0}$ et obtenir le bit $b \in \{0, 1\}$
   (c) si $b = 1$, $\mathcal{N}$ met à jour Ψ avec $\Psi_{x_i=0}$
   si $b = 0$, $\mathcal{N}$ met à jour Ψ avec $\Psi_{x_i=1}$ (la formule booléenne obtenue en remplaçant chaque clause de Ψ par la clause obtenue en fixant la valeur x_i à 1).
3. si $\Psi_{x_n=0}$ est vraie ou $\Psi_{x_n=1}$ est vraie, retourner 1
   sinon retourner 0

Montrer que si la machine de de Turing probabiliste $\mathcal{N}$ prend en entrée une formule booléenne Φ satisfiable, alors la formule booléenne Ψ obtenue à la fin de la i-ème itération de boucle est insatistifiable avec probabilité inférieure ou égale à

$$\sum_{j=1}^{i} \frac{1}{4^{n-j}} = \frac{1}{4^{n-1}} + \cdots + \frac{1}{4^{n-i}}$$

2.d] En déduire, en utilisant la machine de Turing probabiliste $\mathcal{N}$, que $\mathsf{SAT} \in \mathsf{RP}$.

2.e] Conclure.

Exercice 3 : Test d'identités entières

Il a été récemment démontré que l'entier 42 s'écrit sous la forme d'une somme de trois cubes de la façon suivante :

$$42 = (-80538738812075974)^3 + 80435758145817515^3 + 12602123297335631^3 \quad (1)$$

Trouver cette décomposition a demandé une grande puissance de calcul et même si sa vérification est plus simple, elle demande d'écrire des nombres entiers de plus de 50 chiffres décimaux. Dans cet exercice, nous allons reprendre le principe de la « preuve par 9 » sous une forme probabiliste pour tester des identités sur des entiers (de façon similaire au test d'identités polynomiales vu en cours).

3.a] Montrer que la relation (1) est vraie modulo 10.

3.b] Étant donné deux entiers a_1 et a_2 strictement inférieurs en valeur absolue à une borne $B \geq 2$ (c'est-à-dire tels que $|a_1| < B$ et $|a_2| < B$). Montrer que $a_1 = a_2$ si et seulement si il existe des nombres premiers distincts $p_1, \ldots, p_\ell$ tels que

$$a_1 \equiv a_2 \bmod p_i, \forall i \in \{1, \ldots, \ell\} \text{ et } p_1 \cdots p_\ell > 2B$$

3.c] Soient deux entiers a_1 et a_2 représentés sous une forme polynomiale comme dans l'équation (1). Soit $B \geq 2$ une borne supérieure stricte sur a_1 et a_2 en valeur absolue (c'est-à-dire telle que $|a_1| < B$ et $|a_2| < B$). Supposons que $a_1 \neq a_2$.

Montrer qu'il existe une constante $c > 0$ telle que, pour tout entier $n \geq 1$, la probabilité qu'un nombre premier inférieur à 2^n tiré uniformément aléatoirement divise $(a_1 - a_2)$ est inférieure ou égale à $cn \log(B)/2^n$.
Indication. Rappelons une forme faible du théorème des nombres premiers qui affirme qu'il existe une constante $c' > 0$ telle que, pour tout entier $n \geq 1$, le nombre de nombres premiers inférieurs à 2^n, est supérieur ou égal à $c' \cdot 2^n/n$.

3.d] Soit $B \geq 2$. Un entier a avec $|a| < B$ est dit *représentable sous une forme polynomiale utilisant d opérations arithmétiques* (comme dans l'équation (1)) si il existe un polynôme P en n variables et n entiers $\alpha_1, \ldots, \alpha_n$ avec $|\alpha_i| < B$ tels que $a = P(\alpha_1, \ldots, \alpha_n)$ et l'évaluation du polynôme P nécessite au plus d additions d'entiers et d multiplications d'entiers.

Donner un algorithme probabiliste pour tester si deux entiers a_1 et a_2, représentés sous une forme polynomiale utilisant d opérations arithmétiques et avec $|a_1| < B$ et $|a_2| < B$, sont égaux.

Estimer sa probabilité d'erreur et sa complexité.