IFT503/711 – Théorie du calcul

Université de Sherbrooke

Devoir 5

Enseignant:	Manuel Lafond
Date de remise:	jeudi 2 avril 2020 avant 11h59 PM
À réaliser:	individuellement ou à deux au 1er cycle individuellement aux cycles supérieurs
Modalités:	à remettre électroniquement via turnin
Pointage:	sur 40 points au 1er cycle (+ 5pts bonus pour ★) sur 50 points aux cycles supérieurs

Question 1.

(a) Soit $G = (V, E)$ un graphe. Un sous-ensemble $X \subseteq V$ est appelé *dominant* si, pour tout sommet $u \in V \setminus X$, il existe $v \in X$ tel que $uv \in E$. Considérez le problème d'optimisation suivant: 5.5 pts

MIN-k-DOMSET
Entrée: un graphe $G = (V, E)$ dans lequel chaque sommet a exactement k voisins;
Sortie: un ensemble dominant de G de taille minimum.

Montrez qu'il existe une $(k + 1)$-approximation pour le problème MIN-k-DOMSET.

(b) Nous avons vu que MAX-SAT, qui cherche à maximiser le nombre de clauses satisfaites, admet une $\frac{1}{2}$-approximation. Considérez MIN-UNSAT, le problème analogue de minimisation: 5.5 pts

MIN-UNSAT
Entrée: une formule booléenne CNF ϕ;
Sortie: une assignation qui minimise le nombre de clauses de ϕ non-satisfaites.

Montrez que si P $\neq$ NP, il n'existe pas de c-approximation pour MIN-UNSAT pour toute constante $c > 1$.

Question 2.

(a) Montrez que $\text{PCP}(0, 0) = \text{P}$. 2 pts

(b) Montrez que $\text{PCP}(\log n, 0) = \text{P}$ (donc que $\log n$ bits aléatoires n'ajoutent pas de puissance à P). 5 pts

(c) Montrez que s'il existe un vérificateur $(\log(\log n), q)$-PCP pour SAT, où q est une constante, alors P = NP. Notez qu'il n'y a aucune constante qui affecte le $\log \log n$. 8 pts

Boni [5 points] : montrez que si SAT $\in \text{PCP}(\log(\log n), 1)$, alors P = NP. Ici, une constante affecte le $\log \log n$. *Indice* : sur entrée ϕ, considérez les fonctions $V_{r_i}^{w}$ telles que vues en cours. On peut les utiliser pour transformer ϕ en une formule équivalente ϕ' beaucoup plus petite. On peut même répéter avec ϕ'.

Question 3.

(a) Soit α une constante. Un graphe $G = (V, E)$ est appelé α-*dense* si $|E| \geq \alpha |V|^2$. Soit le langage 6 pts

$$\alpha\text{-DENSE-CLIQUE} = \{\langle G, k\rangle : G \text{ est un graphe } \alpha\text{-dense qui contient une clique de taille au moins } k\}$$

Montrez que α-DENSE-CLIQUE est décidable en temps $2^{O(\sqrt{|V|+|E|})} poly(|V|)$.

(b) En supposant que la ETH est vraie, montrez que pour une certaine constante α, il n'existe pas d'algorithme qui décide α-DENSE-CLIQUE en temps $2^{o(\sqrt{|V|+|E|})} poly(|V|)$. 8 pts

En d'autres termes, tout algorithme pour α-DENSE-CLIQUE doit prendre un temps $2^{\Omega(\sqrt{|V|+|E|})} poly(|V|)$.

Rappel: la ETH dit que tout algorithme pour 3-SAT sur n variables et au plus cn clauses doit prendre un temps $2^{\Omega(n)} poly(n)$. Considérez la réduction classique de 3-SAT à CLIQUE.

★ Question 4. (cycles supérieurs)

Dans cette question, nous allons démontrer que MAX-CLIQUE n'admet pas de (n^ε)-approximation pour une certaine constante ε. Rappelons que dans MAX-CLIQUE, on a en entrée un graphe et on cherche la clique de taille maximum.

Un *n-booster* est une collection $\mathcal{S}$ de sous-ensembles de $\{1, \ldots, n\}$ telle que:

— $|\mathcal{S}| \leq n^k$ pour une certaine constante $k \in \mathbb{N}$;

— pour tout $A \subseteq \{1, \ldots, n\}$, on a

$$|\mathcal{S}| \left(\frac{|A|}{n} - \frac{1}{100} \right)^{\log n} \leq |\{X : X \in \mathcal{S} \text{ et } X \subseteq A\}| \leq |\mathcal{S}| \left(\frac{|A|}{n} + \frac{1}{100} \right)^{\log n}$$

i.e. le nombre de sous-ensembles de A dans $\mathcal{S}$ est borné selon la taille de A, et ce pour tous les A.

Il a été démontré qu'il existe un algorithme qui, sur entrée 1^n, construit un n-booster en temps polynomial. Vous pouvez utiliser cet algorithme en boîte noire.

Soit $G = (V, E)$ un graphe où on a étiqueté les sommets $V = \{1, \ldots, n\}$. Soit $\mathcal{S}$ un n-booster. On définit le graphe $H_{G,\mathcal{S}}$ comme suit: les sommets sont les ensembles de $\mathcal{S}$, et on ajoute une arête entre les ensembles S_i et S_j si $S_i \cup S_j$ forme une clique dans G.

(a) Montrez que si k est la taille maximum d'une clique dans G, alors la taille maximum d'une clique dans $H_{G,\mathcal{S}}$ est entre $|\mathcal{S}| \left(\frac{k}{n} - \frac{1}{100} \right)^{\log n}$ et $|\mathcal{S}| \left(\frac{k}{n} + \frac{1}{100} \right)^{\log n}$. 3 pts

(b) On sait qu'il est difficile de distinguer si un graphe G a une clique d'au plus $n/6$ sommets, ou une clique d'au moins $n/3$ sommets. Montrez qu'il est difficile de distinguer si un graphe a une clique d'au plus $f(n)$ sommets, ou une clique d'au moins $n^\varepsilon f(n)$ sommets pour une une certaine fonction f et une certaine constante ε. 7 pts

Suggestion: utilisez $H_{G,\mathcal{S}}$ pour amplifier le "gap". Aussi, notez que $c^{\log n} = n^{\log c}$.