Reconnaissance des graphes triangulés en temps linéaire

Le but de cet exercice est de montrer qu'il existe un algorithme qui teste si un graphe $G = (V, E)$ est triangulé en temps linéaire (c'est-à-dire proportionnel à $|V| + |E|$).

Le graphe est représenté par la donnée, pour chaque sommet $v \in V$, du voisinage de v sous la forme d'une liste $A(v)$. Cette liste est de taille degré(v), donc la taille totale de la représentation est $\sum_{v \in V}(1 + \text{degré}(v)) = |V| + 2|E|$. Sans perte de généralité on suppose que le graphe est connexe.

L'algorithme fonctionne en deux étapes. Étape 1: il construit un ordre $v_1 < v_2 < \cdots < v_n$ ($n = |V|$) des sommets, tel que G est triangulé si et seulement si cet ordre est un ordre d'élimination simplicial. Étape 2: l'algorithme vérifie si l'ordre construit est un ordre d'élimination simplicial ou pas.

> **Étape 1**[1] :
>
> Faire $W := V$ et pour chaque sommet v faire $p(v) := 0$.
> [$p(v)$ est appelé *poids* du sommet v.]
>
> Pour $i = 1, \ldots, |V|$, prendre un sommet w de W de poids maximum, lui donner le nom v_i, le retirer de W, et pour tout sommet x de $N(w) \cap W$ faire $p(x) := p(x) + 1$.

1.1. Montrer que l'étape 1 peut être exécutée en temps linéaire.

1.2. Étant donné l'ordre $<$ construit par l'algorithme, montrer que si u, v, w sont trois sommets tels que $u < v < w$, $uw \in E$ et $uv \notin E$, alors il existe un quatrième sommet x tel que $x < v$, $xv \in E$ et $xw \notin E$.

1.3. Disons qu'une chaîne $C = x_0\text{-}x_1\text{-}\cdots\text{-}x_k$ de sommets de G est *mauvaise* si $k \geq 2$, C est sans corde, il existe un entier $j \in \{1, \ldots, k-1\}$ tel que $x_0 < x_1 < \cdots < x_j$ et $x_k < x_{k-1} < \cdots < x_j$, et $x_0 < x_k < x_1$. Le sommet x_0 est appelé le *minimum* de la chaîne.
On suppose que G est un graphe triangulé. Montrer que s'il existe une mauvaise chaîne $C = x_0\text{-}x_1\text{-}\cdots\text{-}x_k$ (avec la notation ci-dessus) alors il existe une mauvaise chaîne dont le sommet minimum y satisfait $y < x_0$. En déduire que G ne possède pas de mauvaise chaîne.

1.4. Montrer que si G est un graphe triangulé alors $v_1 < v_2 < \cdots < v_n$ est un ordre d'élimination simplicial de G.

[1] Cette méthode s'appelle *Maximum Cardinality Search* (MCS) et possède d'autres applications. Une variante plus complexe est la *Lexicographic Breadth-First Search* (LexBFS), encore plus appliquée.

Étape 2:

Soit $v_1 < v_2 < \cdots < v_n$ l'ordre produit à l'étape 1.

Pour chaque sommet $v \in V$, initialiser une liste $B(v) := \emptyset$.

Pour $j = 2, \ldots, n$, soit v_i le sommet adjacent à v_j tel que $i < j$ et i est maximum. Pour chaque sommet v_h adjacent à v_j tel que $h < i$, ajouter v_h à $B(v_i)$.

Vérification : Pour $i = 1, \ldots, n$, faire: si $B(v_i) \not\subseteq A(v_i)$, retourner le message "*G n'est pas triangulé*" et arrêter.

Retourner le message "*G est triangulé*".

Noter que chaque liste $B(v)$ peut contenir des sommets répétés.

2.1. Montrer que l'étape 2 retourne un message correct pour tout graphe G.

2.2. Montrer que le temps total de construction des listes $B(v)_{v \in V}$ est linéaire.

2.3. Donner une manière d'effectuer la vérification en un temps total linéaire[2].

2.4. Pouvez-vous modifier l'algorithme (en conservant un temps total d'exécution linéaire) de telle sorte que, lorsqu'il retourne le message "*G n'est pas triangulé*", il produise aussi un cycle induit sans corde ? Un plus petit cycle induit sans corde ?

[2] Il vaut mieux éviter d'utiliser la matrice d'adjacence du graphe (matrice $(a_{i,j})_{i,j=1,\ldots,n}$ telle que $a_{i,j} = 1$ si $v_i v_j \in E$ et 0 sinon), car cette matrice est de taille n^2 qui peut être beaucoup plus grande et non-proportionnelle à $|E|$.