TD NUMÉRO 7

Algorithmique

Exercice 1 : Clôture transitive

Le but de cet exercice est de montrer que l'on peut calculer la clôture transitive d'un graphe non-orienté en le même temps que la multiplication de deux matrices booléennes et réciproquement.

Montrer les théorèmes suivants :

Théorème 1 : *Si la clôture transitive d'une matrice $n \times n$ peut être calculée en temps $T(n)$, où $T(n)$ satisfait $T(3n) \leq 27T(n)$, alors il existe une constante c telle que le temps $M(n)$ pour multiplier deux matrices $n \times n$ A et B satisfait $M(n) \leq cT(n)$.*

L'hypothèse sur $T(n)$ est valide car $T(n)$ est au plus $O(n^3)$.

Théorème 2 : *Si le produit de deux matrices $n \times n$ arbitraires peut se faire en temps $M(n)$, où $M(n)$ satisfait $M(2n) \geq 4M(n)$, alors il existe une constante c telle que le temps $T(n)$ pour calculer la clôture transitive d'une matrice arbitraire satisfait $T(n) \leq cM(n)$.*

Exercice 2 : Arbre couvrant minimum - Algorithme de Kruskal

Un *arbre couvrant* est un arbre non orienté qui connecte tous les sommets V d'un graphe non orienté. Si les arêtes ont un poids, on recherche un arbre de poids minimum.

1) Montrer les lemmes suivants :

Lemme 1 : *Soit $G = (V, E)$ un graphe connecté, non orienté et $S = (V, T)$ un arbre couvrant pour G. Alors :*
a) pour tous sommets v_1 et v_2 de V, le chemin entre v_1 et v_2 dans S est unique, et
b) si une arête de $E \setminus T$ est ajoutée à S, alors un cycle se forme.

Lemme 2 : *Soit $G = (V, E)$ un graphe connecté, non orienté et c une fonction de coût sur les arêtes. Soit $\{(V_1, T_1), (V_2, T_2), \ldots (V_k, T_k)\}$ une forêt recouvrante pour G avec $k > 1$. Soit $T = \bigcup_{i=1}^{k} T_i$. Supposons $e = (v, w)$ une arête de poids minimale dans $E \setminus T$ telle que $v \in V_1$ et $w \notin V_1$. Alors il existe un arbre recouvrant pour G qui inclut $T \cup \{e\}$ et est de coût minimal parmi les arbre recouvrant pour G qui inclut T.*

2) En déduire un algorithme et les structures de données nécessaires pour calculer un arbre recouvrant minimum.

3) Calculer alors la complexité de votre algorithme.

Exercice 3 : Composante Fortement Connexe : Le problème 2-SAT

Soit $x_1, \ldots, x_n$ un ensemble de n variables booléennes indexées par les entiers $\leq n$. Un littéral sur cet ensemble de variables est une variable x_i, $i \leq n$, ou sa négation $\neg x_i$. Une 2-clause sur cet ensemble de variable est une disjonction de deux littéraux $\ell \vee \ell'$. Une assignation τ est une fonction de l'ensemble $\{x_1, \ldots, x_n\}$ dans $\{0, 1\}$. La valeur d'un littéral ℓ relativement à une assignation τ, soit $V(\ell, \tau)$, est égale à $\tau(x_i)$ si $\ell = x_i$ et à $1 - \tau(x_i)$ si $\ell = \neg x_i$. La valeur d'une clause c relativement à une assignation τ, soit $V(c, \tau)$, est le maximum des valeurs de ses deux littéraux. Le problème 2-SAT s'énonce de la manière suivante :

Étant donné un entier n et une liste de m 2-clauses sur l'ensemble des variables $\{x_1, \ldots, x_n\}$, déterminer s'il existe une assignation τ qui donne à toutes les clauses de la liste la valeur 1 (problème de décision). Calculer une telle assignation si elle existe (calcul d'une solution).

1) On associe à une liste de 2-clauses sur l'ensemble des variables $\{x_1, \ldots, x_n\}$ un graphe orienté G dont les sommets sont les littéraux sur l'ensemble de variables $\{x_1, \ldots, x_n\}$. Pour chaque clause $\ell \vee \ell'$, on ajoute à G une arc de $\neg\ell$ à ℓ' et un arc de $\neg\ell'$ à ℓ (où on pose $\neg\neg x_i = x_i$).

a) Montrer que l'on peut calculer G en temps $O(m+n)$ où n est le nombre de variables et m le nombre de clauses. Donner un algorithme en pseudo-code ainsi que les structures de données nécessaires.

b) Montrer que s'il existe dans G un chemin allant de u à v, il existe aussi un chemin allant de $\neg v$ à $\neg u$.

2) Soit τ une assignation. On étend à l'ensemble des sommets de G en posant $\tau(\neg x_i) = 1 - \tau(x_i)$. Montrer que $V(c, \tau)$ vaut 1 pour toutes les clauses de la liste donnée si et seulement si τ est croissante sur tout chemin γ du graphe G (ce qui signifie : si $\gamma = u_0, \cdots, u_k$, alors $\tau(u_i) \leq \tau(u_{i+1})$ pour $i = 0, \cdots, k-1$).

3) En utilisant l'algorithme qui détermine les composantes fortement connexes d'un graphe orienté, donner un algorithme qui résout le problème de décision 2-SAT. Montrer sa correction et évaluer sa complexité.

4) Comment calculer une assignation lorsque la réponse au problème de décision est positive ? Évaluer la complexité de l'algorithme proposé.

Exercice 4 : Existence des graphes d'expansion

Définition : *Un (n, d, α, c) OU-concentrateur est un graphe biparti $G(L, R, E)$, avec des ensembles de sommets L et R indépendants, chacun d'eux de cardinal n, tel que :*

1. *Tout sommet de L a un degré au plus d,*
2. *Pour tout sous-ensemble S de sommets de L tel que $|S| \leq \alpha n$, il existe au moins $c|S|$ voisins dans R.*

Montrer le théorème suivant :

Théorème 3 : *Il existe un entier n_0 tel que pour tout $n > n_0$, il existe un $(n, 18, 1/3, 2)$ OU-concentrateur.*