Algorithmique et complexité

TD 5/7 – Théorie de la complexité

Exercices complémentaires

Remarque: Les éléments de correction fournis dans le sujet ne sont pas forcément complets. Ils sont là pour vous guider dans votre travail personnel. Nous vous invitons à rédiger une solution comme vous le feriez à l'examen et, si vous avez des questions, à vous tourner vers votre chargé de TD.

Exercice 1 : SUBSETSUM (examen 2018-2019)

On considère les deux problèmes suivants :

BI-PARTITION
Entrée : Un ensemble E de nombres.
Question : Existe-il un sous-ensemble $F \subseteq E$ tel que $\displaystyle\sum_{x \in F} x = \frac{1}{2} \sum_{x \in E} x$?

SUBSETSUM
Entrée : Un ensemble S de nombres et un nombre t
Question : Existe-il un sous-ensemble $T \subseteq S$ tel que $\displaystyle\sum_{x \in T} x = t$?

Question 1

Montrez que SUBSETSUM est dans NP.

Élements de correction :

Il faut écrire en Python un algorithme **polynomial** pour vérifier une solution T (vous pouvez écrire un algorithme de complexité $\mathcal{O}(|S| * |T|)$ ou en $\mathcal{O}(|T|)$ selon la structure de données que vous utilisez pour T).

Question 2

Sachant que BI-PARTITION est NP-complet, montrez que SUBSETSUM est lui aussi NP-complet.

Élements de correction :

Attention à ne pas vous tromper de sens : il faut prendre une instance E de BI-PARTITION et construire une instance de SUBSETSUM en utilisant $S = E$, $t = \frac{1}{2}\sum_{x \in E} x$. Pensez à vérifier que la transformation de (E) en (S, t) est bien polynomiale (linéaire en $|E|$). Montrez ensuite que si F est une solution au problème BI-PARTITION sur E, alors on peut construire une solution T à SUBSETSUM sur (S, t) et, réciproquement, si T est une solution au problème SUBSETSUM sur (S, t), alors on peut construire F solution à BI-PARTITION sur E (en l'occurrence, $F = T$).

Vous aurez alors montré que BI-PARTITION $<_p$ SUBSETSUM. Or BI-PARTITION est NP-complet. Donc SUBSETSUM est NP-difficile.

Or comme il est aussi NP, il est NP-complet.

Exercice 2 : Ensemble dominant

Un ensemble dominant d'un graphe non orienté $G = (V, E)$ est un sous-ensemble $D \subset V$ tel que tout sommet qui n'appartient pas à D possède au moins un voisin dans D.

On s'intéresse au problème décisionnel : Existe-t-il un ensemble de dominant de taille inférieure ou égale à k ?

Question 1

Formalisez le problème de l'ensemble dominant.

Élements de correction :

Données :
- un graphe $G = (V, E)$ où V est l'ensemble des sommets et $E \subseteq V \times V$ l'ensemble des arêtes de G (E est symétrique).
- $k \in \mathbb{N}$ un entier naturel

Question : Existe-t-il un sous-ensemble $D \subset V$ tel que :
- D est un ensemble dominant de G, i.e. $\forall v \in V \setminus D, \exists u \in D, (u, v) \in E$
- la sous-ensemble D est de taille inférieure ou égale à k.

Question 2

En utilisant le problème de couverture par ensembles (Set Cover), montrez que le problème de décision de l'ensemble dominant est NP-complet.

Indication : Il faut construire un graphe $G = (V, E)$ où :
- $V = I \cup U$
- $E = \underbrace{\{(i,j), i, j \in I \wedge i \neq j\}}_{E_1} \cup \underbrace{\{(i,u), i \in I, u \in S_i\}}_{E_2}$; ainsi (V, E_2) est un graphe biparti, (I, E_1) est une clique et (U, E) un ensemble stable.

Élements de correction :

Vérifier une solution à Dominant est polynomial. Il s'agit de vérifier que chaque noeud du graphe v dans V est bien voisin d'un noeud u dans le dominant D, i.e. $(u, v) \in E$ (on peut utiliser une matrice d'adjacence, donc on effectue $|V|$ fois recherche parmi $|D|$ noeuds, $O(|V||D|)$), donc Dominant est dans NP. (voir l'exo pratique correspondant dans les Notebooks)

La réduction de Set cover à Dominant est clairement polynomiale (quadratique). Il faut ensuite vérifier l'équivalence entre instances positives. Source (Wikipédia) : `https://en.wikipedia.org/wiki/Dominating_set#L-reductions`

- $\Rightarrow$ si $C = \{S_i, i \in D\}$ est une solution au set cover avec $D \subseteq I$, alors D est un dominant dans G : d'abord pour chaque $u \in U$ il y a un $i \in D$ tq $u \in S_i$ et par construction, u et i sont adjacents ; donc u est dominé par i. Et comme I est une clique, chaque $i \in I$ est adjacent à D comme $D \subseteq I$.
- $\Leftarrow$ le sens inverse est important et plus compliqué. Si D est un dominant de G. On a donc $D \subseteq U \cup I$. Il est possible de construire un nouveau dominant X tq. $|X| \leq |D|$ et $X \subseteq I$: simplement en remplaçant chaque $u \in D \cap U$ par un voisin $i \in I$ de u. Du coup, $C = \{S_i, i \in D\}$ est une solution au set Cover avec $|C| = |X| \leq |D|$.

Set Cover est NP-Complet, donc Dominant est NP-Difficile. De plus, vérifier une solution à Dominant est polynomial (est dans NP), donc Dom est NP-complet.

Exercice 3 : Problème d'une chaîne hamiltonienne

Nous allons supposer que le problème Cycle Hamiltonien ci-dessous est NP-Complet :

Données : un graphe **non orienté** $G = (V, E)$

Question : G contient-il un cycle hamiltonien, c'est-à-dire un chemin permettant de visité exactement une fois tous les sommets (sauf le sommet de départ qui est de nouveau visité à l'arrivée) ?

Considérons maintenant le problème Chaine Hamiltonienne suivant :

Données : Un graphe non orienté $G = (V, E)$, deux sommets u et v distincts de G

Question : G contient-il une chaîne hamiltonienne entre u et v ?

Explication : Une chaîne hamiltonienne est un chemin qui passe une seule fois par chaque sommet du graphe. On peut transformer un cycle hamiltonien en chaîne hamiltonienne, en enlevant une arête quelconque du cycle. En revanche, une chaîne hamiltonienne ne peut être étendue en un cycle hamiltonien que si ses deux extrémités sont adjacentes.

Question 1

Montrer que le problème Chaine Hamiltonienne est dans NP.

Élements de correction :

Le problème Chaine Hamiltonienne est dans NP, car étant donnée une chaîne, on peut vérifier en temps polynomial $\mathcal{O}(|V|^2)$ si elle passe une fois et une seule par chaque sommet du graphe. Vous pouvez l'écrire simplement en Python en utilisant une matrice d'adjacence pour représenter le graphe.

Question 2

Montrez que le problème Chaine Hamiltonienne est NP-Complet. Pour cela nous ferons une réduction polynomiale du problème Cycle Hamiltonien (connu comme NP-Complet) vers le problème Chaine Hamiltonienne. Soit $\mathcal{I} = \langle G = (V, E)\rangle$ une instance du problème Cycle Hamiltonien. Maintenant nous allons transformer cette instance en une instance $\mathcal{I}' = \langle G' = (V', E'), u, v\rangle$ du problème Chaine Hamiltonienne de la façon suivante :

- soit u un sommet arbitraire de V
- $V' := V \cup \{v\}$ tel que v est un nouveau sommet n'appartenant pas à V
- $E' := E \cup \{(v, l) : l$ est un voisin de u dans $G\}$

Continuez la preuve. . .

Élements de correction :

Cette transformation peut se faire en temps polynomial (nous avons juste copié le graphe G en rajoutant un sommet et moins de $|E|$ arêtes).

La preuve est faite dans deux sens :

- S'il existe une chaîne hamiltonienne P dans G' entre u et v, alors il existe un cycle hamiltonien C dans G. P est de la forme suivante $P = (u, l_1, \ldots, l_{n-1}, v)$. Le cycle $C = (u, l_1, \ldots, l_{n-1}, u)$ est hamiltonien dans G.
- S'il existe un cycle hamiltonien C dans G, alors il existe une chaîne hamiltonienne P dans G'.
  Sans perte de généralité, on peut commencer le cycle en u ainsi $C = (u, l_1, \ldots, l_{n-1}, u)$. Nous construisons la chaîne $P = (u, l_1, \ldots, l_{n-1}, v)$ dans le graphe G'. Cette chaîne est hamiltonienne : elle passe une fois et une seule fois par chaque sommet de G puis visite v en dernier.

Donc Cycle Hamiltonien $\leq$ Chaine Hamiltonienne et Chaine Hamiltonienne est NP-Complet.

Question 3

Chevaliers de la table ronde : Étant donnés n chevaliers, et connaissant toutes les paires de féroces ennemis parmi eux, est-il possible de les placer autour d'une table circulaire de telle sorte qu'aucune paire de féroces ennemis ne soit côte à côte ? De quel problème s'agit-il et quelle est sa classe de complexité ?

Élements de correction :

Il s'agit d'une application du problème NP-complet du Cycle Hamiltonien (car la table est ronde).