DM 2

La qualité et la propreté de la rédaction seront appréciées lors de la notation.
On pourra utiliser le fait que les problèmes vus en cours ou en TD sont $\mathcal{NP}$-complets, notamment :
– Sat
– 3-Sat
– Clique
– Circuit Hamiltonien
– 3-Color
– Vertex Cover (Couverture de sommets)
– 2-Partition *(avec des entiers strictement positifs)*
– Subset-Sum

Exercice 1. *Plus grand carré de 1*

1. Donner un algorithme de programmation dynamique pour résoudre le problème suivant :

   Entrée : une matrice A de taille $n \times m$ où les coefficients valent 0 ou 1.

   Sortie : la largeur maximum K d'un carré de 1 dans A, ainsi que les coordonnées (I, J) du coin en haut à gauche d'un tel carré (autrement dit pour tout i, j, $I \leq i \leq I + K - 1$, $J \leq j \leq J + K - 1$, $A[i, j] = 1$).

2. Quelle est sa complexité ?

Exercice 2. *Ensemble Indépendant*

Montrer que le problème défini ci-dessous est $\mathcal{NP}$-complet.

Independent Set
Instance : Un graphe $G = (V, E)$ et un entier k.
Question : Existe-t-il un stable de taille k dans G ?

Exercice 3. *Chevaliers de la table ronde*

Modéliser le problème Chevaliers défini ci-dessous, et prouver qu'il est $\mathcal{NP}$-complet.

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 ?

Exercice 4. *Sous Chaine*

Dans un graphe orienté, on dit que $\{x_1, \ldots, x_k\}$ est une chaîne transitive de longueur k si et seulement si pour tout $1 \leq i < j \leq k$, $(x_i, x_j) \in E$. Montrer que le problème suivant est $\mathcal{NP}$-complet.

Sous-chaîne transitive :
Instance : Un graphe orienté $D = (V, E)$.
Question : D contient-il une sous-chaîne transitive de longueur au moins $\left\lfloor \frac{|V|}{2} \right\rfloor$?

Indication : Vous pouvez par exemple effectuer une réduction à partir de 3-SAT. Si on se donne un ensemble de clauses $C_1, ..., C_k$, avec $C_i = (x_i^1 \vee x_i^2 \vee x_i^3)$, on peut construire une instance du problème de sous-chaîne transitive en posant $V = \{C_0\} \bigcup_{1 \leq i \leq k} \{C_i, x_i^1, x_i^2, x_i^3\}$ et en choisissant avec soin les arêtes à mettre dans E.

Tournez svp.

Exercice 5. *La 2-Partition dans tous ses états*

Montrer que toutes les variantes suivantes de 2-Partition sont $\mathcal{NP}$-complètes :

1. 2-Part-Even :
   Instance : Un ensemble V de $2n$ entiers strictement positifs $a_1, a_2, \ldots, a_{2n}$.
   Question : Existe-t-il un sous-ensemble $I \subset [1..2n]$ tel que $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$?

2. 2-Part-Eq :
   Instance : Un ensemble V de $2n$ entiers positifs $a_1, a_2, \ldots, a_{2n}$.
   Question : Existe-t-il un sous-ensemble $I \subset [1..2n]$ tel que $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$ et $|I| = n$.

3. 2-Part-Almost-Altern (raccourci en 2-Part-Altern) :
   Instance : Un ensemble V de $2n$ entiers positifs $a_1, a_2, \ldots, a_{2n}$.
   Question : Existe-t-il un sous-ensemble $I \subset [1..2n]$ tel que $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$, et pour $j = 1..n$, soit on a $a_{2j-1} \in I$ et $a_{2j} \notin I$, soit le contraire ?

4. 3-Part :
   Instance : Un ensemble V de n entiers positifs $a_1, a_2, \ldots, a_n$.
   Question : Existe-t-il trois sous-ensembles I_1, I_2 et I_3 partitionnant $[1..n]$ et tels que $\sum_{i \in I_1} a_i = \sum_{i \in I_2} a_i = \sum_{i \in I_3} a_i$?

Exercice 6. *On s'approche...*

Nous considérons ici le problème de la couverture par sommets : étant donné un graphe non-orienté $G = (V, E)$ et une fonction de poids sur les sommets $w : V \to \mathbb{Q}+$, trouver $S \subseteq V$ de poids minimum qui couvre les arêtes (c'est à dire $\forall\, e \in E, \exists\, s \in S : s \in e$). Nous allons étudier ici une 2-approximation. Pour ce faire nous nous intéressons à un type de fonction de poids particulier : les fonctions de poids par degré. w est une fonction de poids par degré si $\exists c > 0, \forall v \in V : w(v) = c \cdot deg(v)$.

1. Soit $\{G = (V, E), w\}$ une instance du problème telle que w est une fonction de poids par degrés. Montrer que $w(V) \leq 2 \cdot OPT$ où $w(V) = \sum_{v \in V} w(v)$.

On sait donc traiter les instances avec une fonction de poids par degrés. La méthode du mille-feuille consiste alors à décomposer une instance quelconque en une famille d'instances avec fonction de poids par degrés.

2. À chaque étape de la décomposition, on cherche la plus grande fonction de poids par degrés p inférieure à w. Expliquer comment calculer p.

L'algorithme du mille-feuille est le suivant :

Données:
$G = (V, E)$;
une fonction de poids quelconque w;
début
  $t \leftarrow 0$;
  $G_0 \leftarrow G$;
  $w_0 \leftarrow w$;
  tant que $G_t = (V_t, E_t)$ *contient une arête* **faire**
    $D_t \leftarrow \{u \in V_t : deg_t(u) = 0\}$;
    $p_t \leftarrow$ plus grande fonction de poids par degrés inférieure à w_t dans G_t;
    $S_t \leftarrow \{u \in V_t : p_t(u) = w_t(u)\}$;
    $G_{t+1} \leftarrow G_t \setminus (D_t \cup S_t)$;
    $w_{t+1} \leftarrow w_t - p_t$;
    $t \leftarrow t + 1$;
  retourner $C = \bigcup_{k=0}^{t-1} S_k$

---

3. Montrer que l'algorithme termine en temps polynomial.

4. Montrer que l'ensemble C de sommets retournés par l'algorithme est bien une couverture.

5. Pour tout $v \in C$, exprimer $w(v)$ en fonction des poids par degrés p_k. Qu'en est-il pour $v \notin C$?
   Remarque : on pourra poser $p_k(u) = 0$ pour tout sommet $u \notin G_k$.

À partir de maintenant, on notera C^* une couverture par sommets optimale pour l'instance de départ $\{G = (V, E), w\}$.

6. Pour toute étape i de l'algorithme, comparer $p_i(C \cap G_i)$ et $p_i(C^* \cap G_i)$.
   Indication : on remarquera que $C \cap G_i$ et $C^* \cap G_i$ sont deux couvertures par sommets de G_i.

7. Montrer que l'algorithme est une 2-approximation du problème de la couverture par sommet minimum avec des poids arbitraires. Trouver un exemple où l'algorithme renvoie effectivement une solution de poids $2.OPT$.