DM 2

À rendre pour le vendredi 12 décembre 2014 à 10h15 dans le casier de votre chargé(e) de TD .

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 *(avec exactement 3 littéraux par clause)*
- Clique
- Circuit Hamiltonien *(sur un graphe orienté)*
- 3-Color
- Vertex Cover (Couverture de sommets)
- Chemin avec Paires Interdites
- Roue
- Subset-Sum

Exercice 1. *Variantes de 3-SAT*

Montrer la $\mathcal{NP}$-complétude des variantes de 3-SAT suivantes :

1. 3-Sat Nae (*not all equal*)
   Instance : Un ensemble de clauses $C_1, \ldots, C_m$, chacune contenant exactement 3 littéraux.
   Question : Existe-t-il une instantiation des variables telle que chaque clause contient un littéral évalué à vrai et un littéral évalué à faux ?
   Indice : On pourra créer m+1 nouvelles variables, et construire une instance avec 2m clauses.

2. 3-Sat Oit (*one in three*)
   Instance : Un ensemble de clauses $C_1, \ldots, C_m$, chacune contenant exactement 3 littéraux.
   Question : Existe-t-il une instantiation des variables telle que chaque clause contient exactement un littéral évalué à vrai (et donc deux littéraux évalués à faux) ?
   *Indice : Pour chaque clause $a \vee b \vee c$, on construira les clauses $\overline{b} \vee x \vee x'$, $\overline{c} \vee y \vee y'$, et **la clause mystère** (à vous de la trouver).*

Exercice 2. *Hamiltonien Non Orienté*

Montrer que le problème suivant est $\mathcal{NP}$-complet :

Cycle Hamiltonien (Non Orienté)
Instance : $G = (V, E)$ un graphe non orienté.
Question : Existe-t-il un cycle hamiltonien dans G, c'est-à-dire un cycle passant par tous les sommets du graphe une et une seule fois ?

Vous ferez une réduction du problème de circuit hamiltonien sur les graphes orientés.

Indice : On pourra pour chaque noeud v du graphe orienté de départ construire trois noeuds v_1, v_2 et v_3 reliés par les arêtes (v_1, v_2) et (v_2, v_3). À vous de trouver la fin de la réduction.

Exercice 3. *Chevaliers de la table ronde*

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

Chevaliers
Instance : Un ensemble de n chevaliers, ainsi que les paires de féroces ennemis parmi eux
Question : 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. *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** $C = \bigcup_{k=0}^{t-1} S_k$

fin

---

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 $2.OPT$.