TD 03 – Le temps de calcul est une fonction de la taille des entrées

Exercice 1. *Taille des entrées*

1. Donner un algorithme en langage de haut niveau pour décider le problème suivant.
   Primalité ; entrée : un entier $x \in \{0,1\}^*$ codé en binaire ; question : x est-il premier ?
2. Borner le temps d'exécution de l'algorithme proposé en 1.
3. Donner un algorithme en langage de haut niveau pour décider le problème suivant.
   Primalité unaire ; entrée : un entier $x \in \{1\}^*$ codé en unaire ; question : x est-il premier ?
4. Borner le temps d'exécution de l'algorithme proposé en 3.

Exercice 2. *SAT et Verif-SAT*

1. Donner un algorithme pour le problème suivant.
   SAT ; entrée : une formule propositionnelle ϕ ; question : ϕ est-elle satisfaisable ?
2. Borner le temps d'exécution de l'algorithme proposé en 1.
3. Donner un algorithme pour le problème suivant.
   Verif-SAT ; entrée : une formule propositionnelle ϕ et une valuation $v : X \to \{\top, \bot\}$ avec X l'ensemble des variables de ϕ ; question : v satisfait-elle ϕ ?
4. Borner le temps d'exécution de l'algorithme proposé en 3.
5. Donner un algorithme non-déterministe pour le problème **SAT**.
6. Borner le temps d'exécution de l'algorithme proposé en 5.

Exercice 3. *Codages « raisonnables »*

1. Donner un algorithme pour transformer un graphe orienté donné par sa matrice d'adjacence
   $$\left(m_{i,j}\right)_{i\in V, j \in V} \text{ telle que } m_{i,j} = 1 \text{ ssi il existe un arc de } i \text{ vers } j,$$
   en listes d'adjacence
   $$\left(\ell_i\right)_{i\in V} \text{ avec } \ell_i \subseteq V \text{ telles que } j \in \ell_i \text{ ssi il existe un arc de } i \text{ vers } j.$$
2. Donner un algorithme pour faire le contraire.
3. Borner le temps d'exécution de chacun de ces deux algorithmes.

Exercice 4. *Temps déterminisite*

Rappel des définitions :

$$\mathsf{P} = \bigcup_{k\in\mathbb{N}} DTIME(n^k),$$

$$\mathsf{EXP} = \bigcup_{k\in\mathbb{N}} DTIME(n^k).$$

1. Montrer que la classe P est close par union, intersection et complémentaire. C'est-à-dire que si L_1 et L_2 sont des problèmes dans P, alors $L_1 \cup L_2$, $L_1 \cap L_2$ et $\Sigma^* \setminus L_1$ (et $\Sigma^* \setminus L_2$) sont aussi des problèmes dans P, avec Σ l'alphabet sur lequel le problème est défini.
2. Au précédent TD nous avons donné une machine M_{pal} pour le problème du **Palindrome** qui fonctionne en temps n^3+2 sur l'alphabet $\Gamma = \{a, b, k, y, B\}$. Expliquer quelle est l'idée du théorème d'accélération linéaire pour résoudre en temps $(1 + \frac{1}{1000})n + \frac{1}{1000}(n^3+2)$ le problème du **Palindrome**.
3. Donner deux classes de complexité en temps déterministe qui sont séparées par le théorème de hiérarchie.
4. Montrer que le problème **2-SAT** est dans la classe P.
5. Montrer que le problème $L_{2SAT+} = L_{2SAT} \cup \{a01bb, t11wu\}$ est dans P, sans utiliser le lemme de clôture de P par changements finis (on supposera que $0, 1, a, b, t, w, u \in \Sigma$). C'est-à-dire, donner un algorithme polynomial pour ce problème (a priori on devrait utiliser le même argument que pour la preuve générale, mais sur un exemple).
6. Montrer que le problème **Clique** est dans la classe EXP.