TD 01 – Ordre de grandeur, codage, langage et problème

Exercice 1. *Ordres de grandeur (1)*

Soit $f : \mathbb{N} \to \mathbb{N}$ et $g : \mathbb{N} \to \mathbb{N}$, rappeler les définitions des notations suivantes :

1. l'ensemble $\mathcal{O}(f)$,
2. l'ensemble $o(f)$.
3. Classer les fonctions suivantes selon leur comportement asymptotique, c'est-à-dire $f(n) \preceq g(n)$ ssi $f(n) \in \mathcal{O}(g(n))$.

$$\begin{array}{llll} f_1(n) = n^2 + 10 & f_2(n) = \log(n) & f_3(n) = 2^n & f_4(n) = \frac{n^2}{4} \\ f_5(n) = n! & f_6(n) = n\ \log(n) & f_7(n) = n^n & f_8(n) = \sqrt{\log(n)} \end{array}$$

Exercice 2. *Ordres de grandeur (2)*

Pour chacun des énoncés suivants, dire s'il est vrai (donner une preuve) ou faux (donner un contre exemple).

1. Si $f(n) \in \mathcal{O}(g(n))$ et $g(n) \in \mathcal{O}(h(n))$ alors $f(n) \in O(h(n))$.

n désignera la taille de l'entrées donnée aux algorithmes.

2. Un algorithme fonctionnant en temps $\mathcal{O}(2^n)$ est plus lent qu'un algorithme fonctionnant en temps $\mathcal{O}(n^2)$.
3. Un algorithme fonctionnant en temps $\mathcal{O}(2^n)$ est plus lent qu'un algorithme fonctionnant en temps $\Omega(n^2)$.
4. Un algorithme fonctionnant en temps $\mathcal{O}(n)$ est plus rapide qu'un algorithme fonctionnant en temps $\Omega(n^3)$.
5. Un algorithme fonctionnant en temps $\mathcal{O}(\log(n))$ est plus rapide qu'un algorithme fonctionnant en temps $\Omega(n^3)$, sur toutes les entrées.

Exercice 3. *Codage*

Soit le codage suivant des couples de nombres entiers sur un alphabet binaire : $(x, y) \mapsto x_1 0 x_2 0 x_3 0 \ldots x_{n-1} 0 x_n 1 y_1 y_2 \ldots y_m$ pour $x = x_1 x_2 \ldots x_n$ et $y = y_1 y_2 \ldots y_m$ deux nombres entiers en base 2.

1. Donner le codage du couple $(10, 5)$.
2. Donner le couple associé au codage 1010111010.
3. Donner le nombre de bits utilisés pour coder un couple (x, y).

Soit le codage suivant des couples de nombres entiers sur un alphabet binaire : $(x, y) \mapsto t_1 0 t_2 0 \ldots t_\ell 0 t_\ell 1 x_1 x_2 \ldots x_n y_1 y_2 \ldots y_m$ pour $x = x_1 x_2 \ldots x_n$ et $y = y_1 y_2 \ldots y_m$ deux nombres entiers en base 2, et $t = t_1 t_2 \ldots t_\ell$ la taille de x en base 2.

4. Donner le codage du couple $(10, 5)$.
5. Donner le couple associé au codage 100011100011001.

6. Donner le nombre de bits utilisés pour coder un couple (x, y).

On souhaite généraliser ce codage a un ensemble de k entiers $w = (w^1, w^2, \ldots, w^k)$ avec $w^i \in \{0, 1\}^*$ pour $1 \leq i \leq k$.

7. Proposer un tel codage.
8. Donner le codage de l'ensemble $(6, 11, 10, 3)$.
9. Donner le nombre de bits utilisés pour coder un ensemble arbitraire $w = (w^1, w^2, \ldots, w^k)$.

Exercice 4. *Langages et problèmes*

Soit le problème suivant :

SAT
entrée : une formule FNC [1] ϕ.
question : ϕ est elle satisfaisable ?

1. Quel est le langage $\mathcal{L}_{SAT}$ associé à ce problème ? Sur quel alphabet est (par exemple) défini ce langage ?
2. Donner un exemple de mot $x \in \mathcal{L}_{SAT}$.
3. Donner un exemple de mot $x \notin \mathcal{L}_{SAT}$.

Soit le problème suivant :

CLIQUE
entrée : un graphe non-orienté $G = (V, E)$ et un entier k.
question : G contient-il une clique [2] de taille k ?

4. Quel est le langage $\mathcal{L}_{CLIQUE}$ associé à ce problème ? Sur quel alphabet est (par exemple) défini ce langage ?
5. Donner un exemple de mot $x \in \mathcal{L}_{CLIQUE}$.
6. Donner un exemple de mot $x \notin \mathcal{L}_{CLIQUE}$.

Exercice 5. *Problèmes de décision et de calcul*

Soit le problème de calcul suivant :

ENSEMBLE INDEPENDANT (calcul)
entrée : un graphe non-orienté $G = (V, E)$.
question : Quel est la taille du plus grand ensemble indépendant [3] contenu dans G ?

1. Donner le problème de décision associé à ce problème de calcul.
2. Si l'on sait résoudre le problème de décision pour ENSEMBLE INDEPENDANT, expliquer comment résoudre le problème de calcul pour ENSEMBLE INDEPENDANT.
3. Combien d'appels à l'algorithme pour le problème de décision sont réalisés pour obtenir la solution du problème de calcul ?