SÉRIE 9 (Chapitres 10 et 11)

Question # 1
Démontrez que le nombre de déplacements de disques nécessaires pour résoudre le problème des tours de Hanoi (avec n disques) doit être supérieur ou égal à $2^n - 1$.

Question # 2
Un algorithme qui utilise un nombre polynomial de fois un autre algorithme à temps polynomial est-il assurément un algorithme à temps polynomial ?

Question # 3
Un problème p peut être résolu par un algorithme ayant un temps d'exécution $O(n^{\lg(n)})$. Laquelle des affirmations suivantes est vraie ?

A) Le problème est nécessairement traitable.
B) Le problème est nécessairement intraitable.
C) Aucune de ces réponses n'est vraie.

Question # 4
Trouvez un minorant pour un algorithme devant lister tous les nombres premiers contenus dans un vecteur $V[1..n]$.

Question # 5
Trouvez un minorant pour un algorithme devant énumérer tous les mots de passe possibles formés de n caractères choisis parmi $\{a..z, A..Z, 0..9\}$.

Question # 6
Trouvez un minorant pour un algorithme qui doit multiplier une matrice de dimensions $a \times b$ par une matrice de dimensions $b \times c$.

Question # 7
Considérez l'algorithme de force brute pour résoudre le problème de décider si un nombre n est premier ou composé (Algorithme 1). Cet algorithme est-il suffisant pour conclure que le problème mentionné appartient à la classe P ?

Algorithme 1 : ForceBruteEstCompose(n)

pour $i = 2..\lfloor n/2 \rfloor$ **faire**
    si $n \mod i = 0$ **alors**
        retourner *«oui»*
retourner *«non»*

Question # 8
Le problème de partition *se définit de la façon suivante : Soit A un ensemble d'entiers. Existe-t-il un sous-ensemble $B \subset A$ tel que la somme des éléments dans B est égale à la somme des éléments dans A qui ne sont pas dans B ?*

$$\sum_{b \in B} b = \sum_{c \in A \setminus B} c$$

Le problème du sac à dos décision se définit de la façon suivante : Soit n objets avec des poids $w_1, \ldots, w_n$ et des valeurs $v_1, \ldots, v_n$. Soit un sac à dos de capacité maximale W et un nombre V. Existe-t-il un sous ensemble de ces objets tel que le poids total est au plus W et tel que la valeur totale est au moins V ?

A) *Prouvez que le problème du sac à dos version décision appartient à la classe NP.*

B) *Prouvez que le problème de partition est réductible polynomialement au problème de décision pour le problème du sac à dos.*

C) *Sachant que le problème de partition est NP-complet, que pouvez-vous conclure pour le problème du sac à dos version décision ?*

Question # 9

Le problème de satisfiabilité (SAT) se définit de la façon suivante : Soit $L = \{l_1, l_2, \ldots, l_n\}$, un ensemble de littéraux qui peuvent prendre des valeurs booléennes : `vrai` ou `faux`. Soit $C = \{c_1, c_2, \ldots, c_m\}$, un ensemble de clauses où chaque clause est une disjonction de littéraux. Existe-t-il une assignation de valeurs aux littéraux de L qui satisfasse la conjonction des clauses de C ?
Par exemple, soit l'instance suivante :

$$L = \{l_1, l_2, l_3, l_4\}, \tag{1}$$
$$C = \{\{l_1, \neg l_2\}, \{\neg l_1, l_2, \neg l_3, l_4\}, \{\neg l_1, \neg l_3\}, \{l_4\}\}. \tag{2}$$

Cette instance représente le problème de satisfiabilité suivant :

$$(l_1 \vee \neg l_2) \wedge (\neg l_1 \vee l_2 \vee \neg l_3 \vee l_4) \wedge (\neg l_1 \vee \neg l_3) \wedge (l_4). \tag{3}$$

Une solution à cette instance est $l_1 = \texttt{vrai}$, $l_2 = \texttt{vrai}$, $l_3 = \texttt{faux}$, $l_4 = \texttt{vrai}$.
Le problème de satisfiabilité à 3 littéraux (3SAT) est identique au problème SAT, mais avec la contrainte supplémentaire que toute clause $c \in C$ doit porter sur exactement 3 littéraux.
Sachant que SAT est un problème NP-complet, prouvez que 3SAT est NP-complet.

Question # 10

Utilisez la technique du retour arrière pour trouver un cycle hamiltonien dans le graphe de la figure 1.

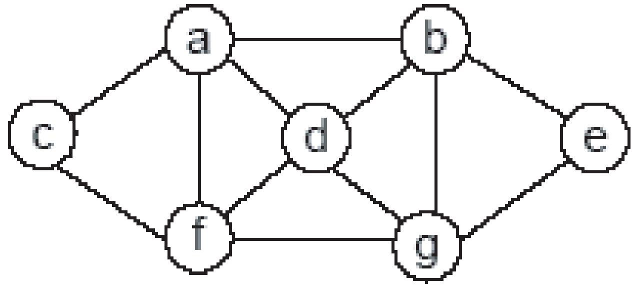

FIGURE 1 – Trouver un cycle hamiltonien sur ce graphe

Question # 11
Utilisez la technique du retour arrière pour générer toutes les permutations de $\{1, 2, 3, 4\}$.

Question # 12
Quelle structure de données utiliseriez-vous pour garder la trace des noeuds actif dans un algorithme de "branch-and-bound" pour la version "best-first" ?

Question # 13

Résolvez l'instance suivante du problème du sac à dos avec l'algorithme de "branch-and-bound" pour $W = 16$ *:*

objet	poids	valeur
1	10	100
2	7	63
3	8	56
4	4	12

Question # 14

A) Pour le problème du sac à dos, élaborez une façon de calculer les bornes qui est meilleure que celle utilisée au Chapitre 11.

B) Utilisez votre nouveau calcul de borne pour résoudre l'instance de la question 13.