
TD 06 – Classes P et NP

Exercice 1.*DTIME()*, *NTIME()*

1. Cocher la bonne réponse.
Une machine de Turing non-déterministe accepte lorsque :
 - tous les chemins d'exécution possibles acceptent.
 - au moins un chemin d'exécution possible accepte.
 - plus de la moitié des chemins d'exécution possibles acceptent.
2. Qu'est-ce que l'ensemble $DTIME(f(n))$?
3. Qu'est-ce que l'ensemble $NTIME(g(n))$?

Exercice 2.

P

1. Donner la définition de P avec des mots.
2. Donner la définition formelle de P.
3. Montrer que le problème suivant est dans P :

Triangle*entrée* : un graphe non-orienté $G = (V, E)$.*question* : le graphe G contient-il un triangle* ?*un triangle est un ensemble de 3 sommets $v_1, v_2, v_3 \in V$ tels que $v_1v_2, v_1v_3, v_2v_3 \in E$.**Exercice 3.**

NP

1. Donner la définition de NP avec des mots.
2. Donner la définition formelle de NP.
3. Montrer que le problème suivant est dans NP :

Somme de sous-ensemble*entrée* : un ensemble d'entiers relatifs $S = \{x_1, x_2, \dots, x_n\}$ avec $x_i \in \mathbb{Z}$ pour $1 \leq i \leq n$.*question* : S contient-il un sous-ensemble d'entiers dont la somme vaut zéro* ?*c'est-à-dire un sous-ensemble $S' \subseteq S$ tel que $\sum_{x \in S'} x = 0$.**Exercice 4.**

1 000 000 \$

Pour chacune des affirmations suivantes, cocher une case pertinente.

1. $P = NP$. vrai faux si je savais le démontrer je gagnerais 1 000 000 \$
2. $P \neq NP$. vrai faux si je savais le démontrer je gagnerais 1 000 000 \$
3. $P \subseteq NP$. vrai faux si je savais le démontrer je gagnerais 1 000 000 \$
4. $NP \subseteq P$. vrai faux si je savais le démontrer je gagnerais 1 000 000 \$

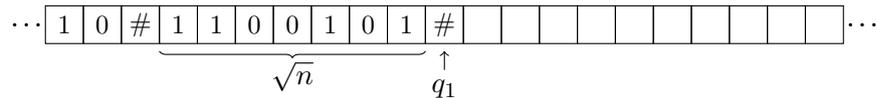
Exercice 5.

Faire deviner une MT non-déterministe

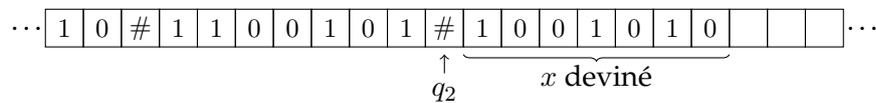
En langage de haut-niveau on utilise l'instruction *deviner* pour écrire les algorithmes non-déterministes. Dans cet exercice nous allons voir comment convertir ces instructions en machines de Turing non-déterministes.

Conventions : dans les schémas les cases vides contiennent des symboles blanc $B \in \Gamma$, et les entiers en binaires sont écrits avec le bit de poids fort à gauche.

1. En imaginant que l'on a un entier $n \in \mathbb{N}$ en entrée dont on veut savoir s'il est premier, convertir en machine de Turing l'instruction *deviner*(un entier $x \in \{1, \dots, \sqrt{n}\}$).
On pourra supposer que le code de machine de Turing pour calculer la racine carrée est déjà écrit, c'est-à-dire que l'on partira de la configuration suivante :

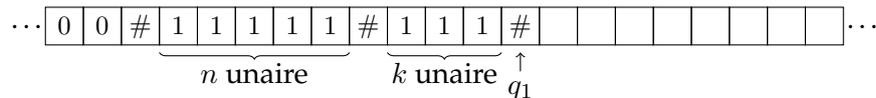


et que l'on veut arriver dans la configuration suivante :

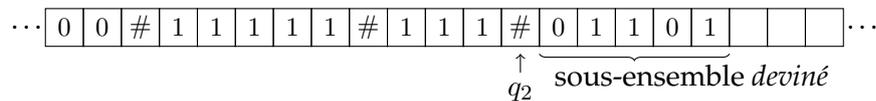


2. En imaginant que l'on a en entrée un graphe non-orienté $G = (V, E)$ à n sommets dont on veut savoir s'il contient une clique de taille k , convertir en machine de Turing l'instruction *deviner*(un sous-ensemble de k sommets de V).

On pourra supposer que l'on part de la configuration suivante :

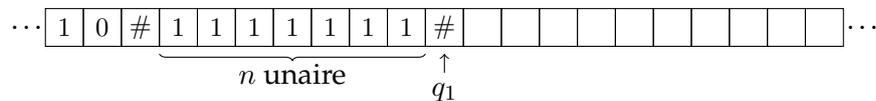


et que l'on veut arriver dans la configuration suivante :



3. En imaginant que l'on a une formule ϕ à n variables (l'ensemble des variables est X avec $|X| = n$) en entrée dont on veut savoir si elle est satisfaisable, convertir en machine de Turing l'instruction *deviner*(une valuation $X \rightarrow \{\perp, \top\}$).

On pourra supposer que l'on part de la configuration suivante :



et que l'on veut arriver dans la configuration suivante :

