

## Algorithmique avancée et Complexité (1h30)

### Exercice 1 : (09 pts) (2,2,2.5,2.5)

1. Déterminer l'invariant des différentes boucles et en déduire l'ordre de complexité des algorithmes suivants :

#### Algorithme A1

**Debut**

$i = N;$

Tant que ( $i > 0$ )

    Faire  $j = 1$  ;

        Tant que ( $j < N$ )

            Faire  $k = 0$  ;

                Tant que ( $k < N$ )

                    Faire  $<->$

$k = k + 2$  ;

                    Fait ;

$j = j * 2$  ;

                Fait ;

$i = i/2$  ;

                Fait ;

**Fin.**

#### Algorithme A2

**Debut**

Pour  $i = 2$  a  $N$

Faire  $p = N$  ;

    Tant que ( $p \geq 1$ )

        Faire

$<->$

$p = p/i$ ;

        Fait ;

    Fait ;

**Fin.**

#### Algorithme A3

Void Recursive ( $G, D$  : entier) ;

**Debut**

    Si ( $D = G$ )

        alors (afficher  $T[1]$ )

    Sinon

        Recursive( $G, D - 1$ )

        Recursive( $G + 1, D$ )

    Fsi ;

**Fin.**

2. Résoudre l'équation de récurrence suivante et en déduire l'ordre de complexité :

$$\begin{cases} T(m, n) = 2 * T\left(\frac{m}{2}, \frac{n}{2}\right) + m * n ; m > 1, n > 1 \text{ et } m \leq n \\ T(m, n) = n \text{ si } m = 1 \\ T(m, n) = m \text{ si } n = 1 \end{cases}$$

### Exercice 2 : (11 pts) (3,4,4)

Etant donné un échiquier de taille  $n*n$ , le problème des  $n$ -Reines consiste à placer  $n$  reines (pions) sur cet échiquier sans que deux reines se menacent mutuellement. Deux reines ou plus se menacent si elles se trouvent sur la même ligne, la même colonne ou bien sur la même ligne diagonale (montante ou descendante). Le problème peut formellement être décrit par une matrice carrée de dimension  $n*n$ , notée  $M$  : Tel que :

$$M[i, j] = \begin{cases} 0 & \text{Si la case est libre} \\ 1 & \text{Si la case est occupé par une reine} \\ -1 & \text{Si la case est menacé par au moins une autre reine} \end{cases}$$

Initialement, tous les cases de la matrice  $M[i, j] = 0$  peuvent accueillir une reine.

Lorsqu'une reine est placée dans une case. la ligne, la colonne et les diagonales qui lui

correspondant sera automatiquement mis à (-1) pour que la reine en question ne soit pas menacée. La matrice gauche montre le placement d'une reine à la première ligne. La matrice droite montre l'état de la matrice suite au placement des 5 reines.

-1	1	-1	-1	-1
-1	-1	-1	0	0
0	-1	0	-1	0
0	-1	0	0	-1
0	-1	0	0	0

-1	1	-1	-1	-1
-1	-1	-1	1	-1
1	-1	-1	-1	-1
-1	-1	1	-1	-1
-1	-1	-1	-1	1

1. Illustrer le processus de résolution sur une matrice  $5 \times 5$ .
2. Proposer un algorithme permettant le placement des  $n$  reines dans une matrice  $n \times n$ .
  - a. Calculer la complexité de l'algorithme proposé.
3. Proposer un algorithme permettant de vérifier, pour une matrice en entrée, si les reines sont correctement placées sans qu'aucune ne puisse attaquer une autre.
  - a. Calculer la complexité de l'algorithme proposé.