Rapport TD1 IA GL

Q1)

$X = X[i][j]$ $i,j \in [1....n]$

$D = X[i][j] \in [1....n]$

$C =$

— $\forall i AllDifrent(X[i][1].....X[i][n]$

— $\forall i AllDifrent(X[1][i].....X[n][i])$

— *square :*

On définit un square comme toutes les cases respectant :

$\forall ij \in [0, ..., n]$

$\forall mq \in [0, ..., \sqrt{n}]$

$m \times \sqrt{n} < i < (m+1) \times \sqrt{n}$

$q \times \sqrt{n} < j < (q+1) \times \sqrt{n}$

$\forall xy AllDiffrent(square[x][y])$

On définit un square comme toutes les cases allant de $m \times \sqrt{n}$ *à* $(m+1) \times \sqrt{n}$ *sur les lignes et les colonnes et ou toutes ces cases sont diférentes.*

Q2)

Le nombre de solution possible dans un sudoku sans contrainte est de $= n^{n^2}$

Q3)

- *1^er Itération*
  - $a = X_{1,2}$
  - $v = 1$
  - $I = I \cup < X_i, v >$
  - *return true*
- *2^es Itération*
  - $a = X_{2,1}$
  - $v = 3$
  - $I = I \cup < X_i, v >$
  - *return true*
- *3^es Itération*
  - $a = X_{2,3}$
  - $v = 1$
  - $I = I \cup < X_i, v >$
  - *return true*
- *4^es Itération*
  - $a = X_{3,1}$
  - $v = 2$
  - $I = I \cup < X_i, v >$
  - *return true*
- *5^es Itération*
  - $a = X_{3,3}$
  - $v = 4$
  - $I = I \cup < X_i, v >$
  - *return true*
- *6^es Itération*
  - $a = X_{3,4}$
  - $v = 1$
  - $I = I \cup < X_i, v >$
  - *return true*
- *7^es Itération*
  - $a = X_{4,2}$
  - $v = 4$
  - $I = I \cup < X_i, v >$

- *return true*

— 8^{es} *Itération*

- $a = X_{4,3}$
- $v = 3$
- $I = I \cup < X_i, v >$
- *return true*

4	1	2	3
3	2	1	4
2	3	4	1
1	4	3	2

Q4)

Je déroule la fonction revise que pour la première itération.

— 1^{er} *Itération*

— $D[X_{1,2}] = \{1,2,3,4\}$

- $v_i = 1$
- $v_j = 2$
- $D[X_{1,2}] = \{1,2,3,4\}$
- $v_i = 2$
- $v_j = 2$
- $D[X_{1,2}] = \{1,3,4\}$
- $v_i = 3$
- $v_j = 2$
- $v_j = 3$
- $D[X_{1,2}] = \{1,4\}$
- $v_i = 1$
- $v_j = 2$
- $v_j = 3$

- $v_j = 4$
- $D[X_{1,2}] = \{1\}$

— 2^{es} *Itération*

- $D[X_{2,1}] = \{1\}$ *après avoir exécuter revise*(X_{21}, C_{ij})

— 3^{es} *Itération*

- $D[X_{2,3}] = \{1,3\}$ *après avoir exécuter revise*(X_{23}, C_{ij}) *or* $D[X_{2,1}] = \{1\}$ *donc* $D[X_{2,3}] = \{3\}$

— 4^{es} *Itération*

- $D[X_{3,1}] = \{4\}$ *après avoir exécuter revise*(X_{31}, C_{ij})

— 5^{es} *Itération*

- $D[X_{3,3}] = \{1,4\}$ *après avoir exécuter revise*$(X3_3, C_{ij})$ *or* $D[X_{3,4}] = \{1\}$ *donc* $D[X_{2,3}] = \{4\}$

— 6^{es} *Itération*

- $D[X_{3,4}] = \{1\}$ *après avoir exécuter revise*(X_{34}, C_{ij})

— 7^{es} *Itération*

- $D[X_{4,2}] = \{4\}$ *après avoir exécuter revise*(X_{42}, C_{ij})

— 8^{es} *Itération*

- $D[X_{4,3}] = \{3,4\}$ *après avoir exécuter revise*(X_{43}, C_{ij}), *or* $D(X2_{,3}) = \{3\}$ *donc* $D[X_{4,3}] = \{4\}$

4	1	2	3
3	2	1	4
2	3	4	1
1	4	3	2

Q5)

Algorithm 1 AllBactrack(<X, D, C>, I)

```
if I is complete then
    return FALSE
end if
Select a variable X_i not in I
for each: v in D(X_i) do
if I ∪ <X_i,v> is locally consistent then
    if AllBactrack(<X, D, C>, I ∪ <X_i,v>) then
        return TRUE
    end if
end if

return FALSE
```

Rapport TP1 IA GL

1 Utilisation CLI

Le programme comporte une CLI qui premet de naviguer entre les différents exercice.

```
0) Exit
1) Question 6 résultat backtrack pour un sudoku 4*4
2) Question 7 Test unitaire de PPC et du backtrack pour une solution (peut-être assez long)
3) Question 8 Afficher toutes les solutions du sudoku 4*4 en PPC
4) Question 8 Afficher toutes les solutions du sudoku 4*4 en BT
5) Question 9 Test unitaire de PPC et du backtrack pour toutes les solutions (peut-être assez long)
6) Question 10 résultat de PPC sur la figure 2
7) Question 11 résultat de PPC sur la figure 3
8) Question 12 résultat de PPC sur la figure 4
9)  résultat de PPC sur une figure passé en paramètre
10) résultat de PPC sur une figure Greater Than Sudoku passé en paramètre
11) Afficher tous les exercices d'un coup (peut-être assez long)

Selectionner l'exercice à exécuter :
1

[1, 2, 3, 4]
[3, 4, 1, 2]
[2, 1, 4, 3]
[4, 3, 2, 1]
```

Pour exécuter un exercice il suffit de taper le numéro marqué devant l'exercice à exécuter.
Les exercices sur les tests unitaires peuvent être assez long à exécuter.
Pour les options 9 et 10 de la CLI il faut passer en paramètre le lien de la figure

```
Selectionner l'exercice à exécuter :
10
Donner le lien du fichier csv représentant la figure Greater Than Sudoku
C:\Users\matth\IdeaProjects\TP1_IA\out\artifacts\TP1_IA_jar\src\main\Figure\Figure4.csv
┌─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┬─────┐
│ 2 < 6 < 7 │ 1 < 3 < 9 │ 8 > 5 > 4 │
├─v─┼─^─┼─v─┼─^─┼─^─┼─v─┼─v─┼─^─┼─v─┤
│ 1 < 9 > 3 │ 5 > 4 < 8 │ 7 > 6 > 2 │
├─^─┼─v─┼─^─┼─^─┼─^─┼─v─┼─v─┼─^─┼─v─┤
│ 8 > 5 > 4 │ 6 < 7 > 2 │ 3 < 9 > 1 │
├───┼───┼───┼───┼───┼───┼───┼───┼───┤
│ 9 > 3 < 8 │ 2 > 1 < 5 │ 6 > 4 < 7 │
├─v─┼─v─┼─v─┼─^─┼─^─┼─v─┼─v─┼─v─┼─^─┤
│ 6 > 2 > 1 │ 7 < 9 > 4 │ 5 > 3 < 8 │
├─v─┼─^─┼─^─┼─v─┼─v─┼─^─┼─v─┼─v─┼─^─┤
│ 4 < 7 > 5 │ 3 < 8 > 6 │ 2 > 1 < 9 │
├───┼───┼───┼───┼───┼───┼───┼───┼───┤
│ 5 > 4 < 9 │ 8 > 2 < 3 │ 1 < 7 > 6 │
├─^─┼─^─┼─v─┼─v─┼─^─┼─v─┼─^─┼─v─┼─v─┤
│ 7 < 8 > 6 │ 4 < 5 > 1 │ 9 > 2 < 3 │
├─v─┼─v─┼─v─┼─^─┼─^─┼─^─┼─v─┼─^─┼─^─┤
│ 3 < 1 > 2 │ 9 > 6 < 7 │ 4 < 8 > 5 │
└─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┴─────┘
```

2 Test Unitaire

```
BT prend plus de temps pour un Sudoku de taille 4 avec comme temps :
PPC : 3.0
BT : 7096.0
```

On peut voir que sur un Sudoku 4×4 le programme PPC est nettement plus rapide que le BT. On peut donc en conclure que la porgammation par contrainte est plus efficace que la méthode Bactrack.

Pour trouver toutes les solutions d'un Sudoku 4×4 l'algorithme par contrainte est toujours plus rapide que le bactrack.

```
BT prend plus de temps pour un Sudoku de taille 4 avec comme temps :
PPC : 81.0
BT : 71325.0
```

J'ai créé d'autre test unitaire qui test les différents algorithmes sur des sudokus de taille 9 et 16 mais je ne les exécutes pas car le temps d'exécution est trop long pour le bactrack.

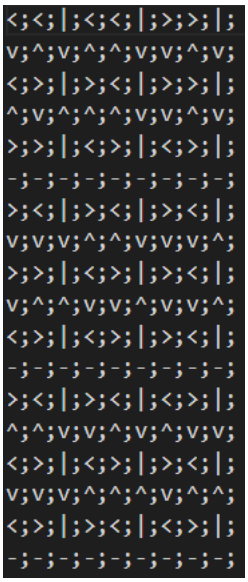
2.1 Explication format fichier

Pour les contraintes du Sudoku le format ressemble à : Les valeurs case avec la valeur

```
8;0;0;0;0;0;0;0;0
0;0;3;6;0;0;0;0;0
0;7;0;0;9;0;2;0;0
0;5;0;0;0;7;0;0;0
0;0;0;0;4;5;7;0;0
0;0;0;1;0;0;0;3;0
0;0;1;0;0;0;0;6;8
0;0;8;5;0;0;0;1;0
0;9;0;0;0;4;0;0
```

0 représentes les cases vides du sudoku celle qui ne subiront pas de contrainte.
Les cases d'une même ligne sont délimité par des ";" et chaque colonne est délimiter par un retour à la ligne.
les valeurs peuvent aller jusqu'a G au-delà les valeurs doivent être noté 17-18-19....

Pour Greater Than Sudoku le format du fichier est représenté comme ceci : Le carac-



tère < signifie que la case avant le < doit-être inférieur à la case suivante inversement pour le caractère >.
Le caractère | est neutre cela signifie qu'il n'y a pas de contrainte appliqué au case à coté de ce caractère.
Le caractère ˆsignifie que la case au-dessus doit-être inférieur à la case du dessous et inversement pour v
Le caractère - est neutre cela signifie qu'il n'y a pas de contrainte applicqué au case au dessus et en dessous de ce caractère.

3 Explication progamme

J'ai créé une fonction Allsolve dans SudokuPPC qui permet d'afficher toutes les solutions d'un sudoku.

```
1                public void Allsolve()
2
```

J'ai modifier la fonction printGrid pour qu'elle puisse aussi afficher les sudokuGT avec mes < et > à chaque case.

```
1                public void printGrid()
2
```

la fonction AddContrainte permet a partir d'un fichier CSV de rajouter des contraintes au Sudoku et de résoudre le sudoku en fonction de ces contraintes. Cette fonction ajuste la taille du sudoku en fonction du fichier CSV.

```
1            public void AddContrainte(String path)
2
```

La fonction ReadCSVContrainte permet de lire un CSV avec des contraintes et rend le sudoku sous forme de ArrayList de ArrayList.

```
1        public ArrayList<ArrayList<Integer>> ReadCSVContrainte(String
     path)
2
```

La fonction SudokuGT permet en fonction d'un CSV passé en paramètre de résoudre un Greater Than Sudoku.

```
1        public void SudokuGT(String path)
2
```

La fonction ReadCSVContrainteGT permet de lire un fichier CSV d'un Greater Than Sudoku et rend une ArrayList de ArrayList représentant tout les $<$ $>$ $\vee$ et $\wedge$ du Sudoku.

```
1        public ArrayList<ArrayList<Character>> ReadCSVContrainteGT(String
     path)
2
```

J'ai créé un fichier java TestSudoku qui contient toutes les fonctions de test de SudokuPPC et SudokuBT.

Les fonctions suivantes permettent respectivement de calculer le temps de résolution d'une solution d'un sudoku de taille 4, 9 puis 16. mais le 9 et 16 n'est pas exécuté car très long.

```
1            public void TimeForSize4()
2            public void TimeForSize9()
3            public void TimeForSize16()
```

```
4
```

Les fonctions suivantes permettent respectivement de calculer le temps de résolution de toutes les solutions d'un sudoku de taille 4, 9. mais le 9 n'est pas exécuté car très long.

```
1               public void AllSolutionTimeForSize4()
2               public void AllSolutionTimeForSize9()
3               public void TimeForSize16()
4
```

J'ai aussi créé un fichier InterfaceCLI qui permet d'exécuter la CLI.