

Projet Sudoku : ESGF 5BD

Membre du groupe :

AIDARA Oumoul khairy

BALDE Mame Fatou

BARO Aminata

[Solution Technique Humaine by MAMEFAT · Pull Request #13 · jsboige/5ESGF-BD-2023 \(github.com\)](#)

Les méthodes que nous avons utilisées nous permettent de résoudre un Sudoku en utilisant l'algorithme de récursion.

La méthode « ResoudreSudoku » parcourt toutes les cases vides du Sudoku et tente d'y placer chaque chiffre possible (de 1 à 9). Si un chiffre est valide, il est placé dans la case, puis la méthode est appelée de manière récursive pour la case suivante. Si le Sudoku est résolu, la méthode renvoie true. Sinon, elle renvoie false.

```
2 références | mamef, Il y a 6 heures | 1 auteur, 1 modification
static bool ResoudreSudoku(int[,] sudoku)
{
    int ligne = 0;
    int colonne = 0;

    if (!TrouverCaseVide(sudoku, ref ligne, ref colonne))
    {
        return true; // Sudoku résolu
    }

    for (int chiffre = 1; chiffre <= 9; chiffre++)
    {
        if (EstValide(sudoku,
            ligne, colonne, chiffre))
        {
            sudoku[ligne, colonne] = chiffre;

            if (ResoudreSudoku(sudoku))
            {
                return true;
            }
        }
    }
}
```

```
        sudoku[ligne, colonne] = 0; // Réinitialiser la case
    }

    return false; // Aucune solution trouvée
}
```

La méthode « TrouverCaseVide » recherche la première case vide dans le Sudoku et renvoie ses coordonnées (ligne et colonne). Si aucune case vide n'est trouvée, la méthode renvoie false.

```

1 référence | mamef, il y a 6 heures | 1 auteur, 1 modification
static bool TrouverCaseVide(int[,] sudoku, ref int ligne, ref int colonne)
{
    for (ligne = 0; ligne < 9; ligne++)
    {
        for (colonne = 0; colonne < 9; colonne++)
        {
            if (sudoku[ligne, colonne] == 0)
            {
                return true;
            }
        }
    }

    return false; // Aucune case vide trouvée
}

```

La méthode « EstValide » vérifie si un chiffre donné est valide pour une case donnée. Elle commence par vérifier si le chiffre est déjà présent dans la ligne ou la colonne de la case, puis elle vérifie si le chiffre est déjà présent dans la région 3x3 de la case.

```

1 référence | mamef, il y a 6 heures | 1 auteur, 1 modification
static bool EstValide(int[,] sudoku, int ligne, int colonne, int chiffre)
{
    // Vérifier la ligne
    for (int i = 0; i < 9; i++)
    {
        if (sudoku[ligne, i] == chiffre)
        {
            return false;
        }
    }

    // Vérifier la colonne
    for (int i = 0; i < 9; i++)
    {
        if (sudoku[i, colonne] == chiffre)
        {
            return false;
        }
    }
}

```

```

// Vérifier la région 3x3
int regionLigne = ligne - ligne % 3;
int regionColonne = colonne - colonne % 3;

for (int i = regionLigne; i < regionLigne + 3; i++)
{
    for (int j = regionColonne; j < regionColonne + 3; j++)
    {
        if (sudoku[i, j] == chiffre)
        {
            return false;
        }
    }
}

return true; // Le chiffre est valide
}

```

La méthode « AfficherSudoku » permet d'afficher le Sudoku dans la console.

```
2 références | mamef, Il y a 6 heures | 1 auteur, 1 modification
static void AfficherSudoku(int[,] sudoku)
{
    for (int ligne
        = 0; ligne < 9; ligne++)
    {
        for (int colonne = 0; colonne < 9; colonne++)
        {
            Console.Write(sudoku[ligne, colonne] + " ");
        }
        Console.WriteLine();
    }
}
```

Après exécution du code on obtient le résultat suivant :

```
Sudoku avant résolution :
5 3 0 0 7 0 0 0 0
6 0 0 1 9 5 0 0 0
0 9 8 0 0 0 0 6 0
8 0 0 0 6 0 0 0 3
4 0 0 8 0 3 0 0 1
7 0 0 0 2 0 0 0 6
0 6 0 0 0 0 2 8 0
0 0 0 4 1 9 0 0 5
0 0 0 0 8 0 0 7 9
Sudoku après résolution :
5 3 4 6 7 8 9 1 2
6 7 2 1 9 5 3 4 8
1 9 8 3 4 2 5 6 7
8 5 9 7 6 1 4 2 3
4 2 6 8 5 3 7 9 1
7 1 3 9 2 4 8 5 6
9 6 1 5 3 7 2 8 4
2 8 7 4 1 9 6 3 5
3 4 5 2 8 6 1 7 9
```