

Saé S1.02 : comparaison d'approches algorithmiques

Résolution automatique de Sudoku

Consignes pour la partie 1

Premier programme

Écrivez un premier programme, appelé **ELIMINATION_Version1.c** basé sur l'algorithme vu en séance de Ressource-SAE.

Vous commencerez par écrire un programme basique qui implémentera uniquement la première technique ("singleton nu"). Testez ce programme avec les différentes grilles fournies sur Moodle.

Puis ajoutez progressivement à votre programme d'autres techniques afin de le rendre de plus en plus performant. Testez-le au fur et à mesure avec les grilles fournies.

Ce premier programme utilisera la première structure de données pour les cases d'une grille de Sudoku, la structure **tCase1** rappelée ici :

```
#define N 3
#define TAILLE (N*N)

typedef struct {
    int valeur;
    int candidats[TAILLE];
    int nbCandidats;
} tCase1;

typedef tCase1 tGrille[TAILLE][TAILLE];
```

Deuxième programme

Écrivez une deuxième version de votre programme, appelé **ELIMINATION_Version2.c** qui utilisera la deuxième structure de données **tCase2** :

```
#define N 3
#define TAILLE (N*N)

typedef struct {
    int valeur;
    bool candidats[TAILLE + 1];
    int nbCandidats;
} tCase2;

typedef tCase2 tGrille[TAILLE][TAILLE];
```

Testez ce deuxième programme avec les grilles fournies sur Moodle.

Travaux à déposer

Vous déposerez au plus tard le 7 janvier 2024 avant minuit :

- votre premier programme : **ELIMINATION_Version1.c**, utilisant la structure `tCase1`
- votre deuxième programme : **ELIMINATION_Version2.c**, utilisant la structure `tCase2`
- les captures d'écran des résultats affichés par votre programme pour chaque grille fournie (*cf.* fichiers `.sud`). Les résultats devront se présenter sous cette forme :

```
*****  RESULTATS POUR Grille99.sud  *****
Nombre de cases remplies = 1 sur 58      Taux de remplissage = 1.724 %
Nombre de candidats elimines = 6        Pourcentage d'elimination = 2.597 %
```

Saé S1.02 : comparaison d'approches algorithmiques

Résolution automatique de Sudoku

Consignes pour la partie 2

Première programme : RESOLUTION_1

Vous devez implémenter en C l'algorithme de backtracking vu en séance de Ressource-SAE, en respectant les consignes suivantes :

- votre programme devra gérer des **grilles 4x4** (constante N=4).
- votre programme devra :
 - lire un fichier de grille de sudoku 4x4¹,
 - afficher la grille initiale,
 - résoudre la grille,
 - afficher la grille finale (grille complète),
 - afficher le temps CPU de l'étape de résolution.
- vous pourrez bien sûr reprendre des fonctions et procédures de vos programmes antérieurs (pour l'affichage, la vérification, etc.).
- vous testerez ce programme RESOLUTION_1 avec les grilles 4x4 disponibles sur Moodle¹.

Deuxième programme : RESOLUTION_2

S'il résout une grille de Sudoku à coup sûr, le programme de backtracking est par contre très lent. Réfléchissez à des améliorations possibles et proposez une nouvelle version de votre programme qui soit plus rapide.

Vous testerez ce nouveau programme RESOLUTION_2 avec les grilles 4x4 disponibles sur Moodle et vous comparerez avec les résultats obtenus précédemment.

¹ voir sur Moodle les fichiers *MaxiGrilleA.sud*, *MaxiGrilleB.sud*, etc.

Travaux à déposer

1 - Vous déposerez au plus tard le 17 janvier 2024 avant minuit :

- le code source de votre programme de RESOLUTION-1;
- des captures d'écran montrant le temps CPU nécessaire à votre programme pour résoudre chacune des grilles 4x4 fournies. L'affichage prendra cette forme :

```
Grille initiale
+-----+-----+-----+-----+
| . 6 3 . | 5 13 . 9 | 16 . 1 2 | . 12 15 . |
| 2 16 . . | . 12 . . | . . 8 . | . . 11 5 |
| 13 . 8 5 | 11 3 . . | . . 7 14 | 9 2 . 16 |
| . . 7 12 | . . 2 . | . 5 . . | 10 1 . . |
+-----+-----+-----+-----+
| 3 . 9 . | . 7 8 1 | 13 4 2 . | . 5 . 6 |
| 8 12 4 . | 9 . . . | . . . 6 | . 3 16 10 |
| . . . 2 | 3 . 16 . | . 11 . 8 | 15 . . . |
| 7 . . . | 6 . . . | . . . 16 | . . . 8 |
+-----+-----+-----+-----+
| 6 . . . | 13 . . . | . . . 5 | . . . 15 |
| . . . 3 | 7 . 5 . | . 14 . 15 | 6 . . . |
| 12 1 13 . | 2 . . . | . . . 7 | . 8 10 9 |
| 15 . 5 . | . 16 11 6 | 8 12 10 . | . 4 . 14 |
+-----+-----+-----+-----+
| . . 15 6 | . . 9 . | . 16 . . | 4 14 . . |
| 11 . 14 4 | 8 6 . . | . . 5 12 | 16 10 . 1 |
| 1 9 . . | . 15 . . | . . 3 . | . . 5 11 |
| . 3 12 . | 4 11 . 5 | 7 . 6 1 | . 15 8 . |
+-----+-----+-----+-----+
Grille finale
+-----+-----+-----+-----+
| 14 6 3 11 | 5 13 7 9 | 16 10 1 2 | 8 12 15 4 |
| 2 16 1 9 | 10 12 6 15 | 4 13 8 3 | 14 7 11 5 |
| 13 10 8 5 | 11 3 1 4 | 12 15 7 14 | 9 2 6 16 |
| 4 15 7 12 | 14 8 2 16 | 6 5 9 11 | 10 1 3 13 |
+-----+-----+-----+-----+
| 3 14 9 16 | 15 7 8 1 | 13 4 2 10 | 11 5 12 6 |
| 8 12 4 1 | 9 14 13 11 | 5 7 15 6 | 2 3 16 10 |
| 10 13 6 2 | 3 5 16 12 | 1 11 14 8 | 15 9 4 7 |
| 7 5 11 15 | 6 2 4 10 | 9 3 12 16 | 1 13 14 8 |
+-----+-----+-----+-----+
| 6 11 10 8 | 13 9 12 14 | 3 1 4 5 | 7 16 2 15 |
| 9 4 16 3 | 7 10 5 8 | 2 14 13 15 | 6 11 1 12 |
| 12 1 13 14 | 2 4 15 3 | 11 6 16 7 | 5 8 10 9 |
| 15 2 5 7 | 1 16 11 6 | 8 12 10 9 | 3 4 13 14 |
+-----+-----+-----+-----+
| 5 8 15 6 | 12 1 9 2 | 10 16 11 13 | 4 14 7 3 |
| 11 7 14 4 | 8 6 3 13 | 15 2 5 12 | 16 10 9 1 |
| 1 9 2 13 | 16 15 10 7 | 14 8 3 4 | 12 6 5 11 |
| 16 3 12 10 | 4 11 14 5 | 7 9 6 1 | 13 15 8 2 |
+-----+-----+-----+-----+
Resolution de la grille en 66.599998 sec
```

temps CPU affiché
avec 6 décimales

2 - Vous déposerez au plus tard le 28 janvier 2024 avant minuit :

- le code source de votre programme de RESOLUTION-2. Ce nouveau code source devra comporter en commentaire d'entête des explications courtes mais suffisantes sur les améliorations de cette deuxième version du programme.
- comme précédemment, des captures d'écran montrant le temps CPU nécessaire à votre programme pour résoudre chacune des grilles 4x4 fournies (reprendre l'affichage utilisé précédemment).