Solveur Sudoku
Projet Informatique IN104

Vollhardt Ugo - ugo.vollhardt@centralesupelec.fr

1 Présentation

Chaque binôme aura à implémenter entièrement un solveur de Sudoku, qui devra inclure un certain nombre d'algorithmes de base nécessaires à la résolution.

Pour rappel, un Sudoku est un jeu mathématique qui consiste à remplir une grille de 9x9 cases avec des chiffres allant de 1 à 9, tout en respectant un lot de règles très simples :

- Chaque chiffre ne doit figurer qu'une seule fois sur chaque ligne de la grille.
- Chaque chiffre ne doit figurer qu'une seule fois sur chaque colonne de la grille.
- Chaque chiffre ne doit figurer qu'une seule fois dans chacune des sous sections de 9 cases

Il existe de nombreux algorithmes dédiés à la résolution de tels problèmes, cependant seuls quelques uns seront présentés durant les séances de TP. Les binômes sont néanmoins encouragés à ajouter des algorithmes pour réduire le temps de calcul ou le nombre d'itérations nécessaires à la résolution d'une grille.

Ce projet pourra aussi bien être développé en Python qu'en C, le choix du langage est à la discrétion du binôme. Cependant les problématiques seront légèrement différentes suivant le langage (notamment pour les structures de données utilisées), ainsi le choix est important et surtout définitif.

2 Objectif

L'objectif, comme présenté, est de créer de toute pièce et entièrement, un programme permettant la résolution n'importe quelle grille de Sudoku,

qu'elle que soit sa difficulté.

Ainsi le projet devra contenir les fonctionnalités suivantes :
— Récupération de la description d'une grille de Sudoku depuis une source quelconque (Entrée utilisateur, lecture de fichier, etc.)
— Stockage de la grille dans une structure de donnée adéquate, qui permet une manipulation facile.
— Premier algorithme simple qui permet de résoudre les grilles de difficulté basse.
— Second algorithme, qui consiste à forcer une des valeurs possibles sur une des cases lorsqu'il n'est plus possible de progresser avec l'algorithme simple.
— Une interface qui permette de visualiser le contenu d'une grille (dans la console ou avec une vraie interface graphique si le temps le permet en utilisant GTK+)

En complément pour les binômes qui auront le temps :
— Estimation de la difficulté d'une grille ;
— Génération de grilles aléatoires d'une difficulté donnée ;
— Implémentation d'algorithmes supplémentaires pour améliorer les temps de calcul et le nombre d'itérations ;
— Amélioration de l'interface.

3 Contenu prévisionnel des TPs

Planning à titre indicatif (évoluera en fonction de l'avancement du groupe et des besoins) :

1. Création du dossier Git et implémentation de base du programme (récupération grille et algorithme simple de résolution).
2. Poursuite de l'implémentation de la base et ajout de l'algorithme de forçage, avec rappel des notions de récursivité si besoin.
3. Fin de l'implémentation des deux algorithmes de base.
4. Prise en main de GTK+ pour la création d'une interface graphique.
5. Dernières mises au point et approfondissements.

4 Évaluation

Atteinte des objectifs Pour valider le projet, chaque binôme devra avoir développé un programme contenant les fonctionnalités citées précédemment.

Ainsi chaque binôme devra avoir un programme qui soit capable de lire une grille de Sudoku, la résoudre en utilisant au moins les deux algorithmes présentés durant les séances, et intégrer l'ensemble de ces fonctionnalités au sein d'une interface graphique minimale (afficher une grille à l'écran, pas besoin de plus).
Toute amélioration sera bien entendu valorisée.

Performances & temps de calcul Le but d'un solveur est de résoudre un problème de manière rapide et efficace, ainsi ces deux critères seront aussi pris en compte dans la notation. Les deux algorithmes basiques permettent de résoudre dans un temps et en un nombre d'itérations données le problème, et c'est sur cette base que les points seront attribués.

Robustesse du Code Il y a relativement peu de cas de bord dans la résolution d'un Sudoku, cependant il faut qu'un programme soit capable de fonctionner dans tous les cas, qu'importe ce que fait l'utilisateur avec. Ainsi il est à votre charge de vous en assurer. (Rappel de la loi de Murphy : Tout ce qui peut arriver, arrivera !)

Clarté du code & Commentaires Un code n'est pas seulement pour vous, mais aussi pour celui qui le lit, celui qui le maintient, celui qui passe après vous, ou même pour vous 5 ans après. Ainsi le code devra être clair et explicite, avec ce qu'il faut de commentaires pour expliquer le fonctionnement de vos fonctions et les choix d'implémentation.