TD 14 Sudoku

Lycée Louis Thuillier - Physique - PCSIB

Une grille de Sudoku est une grille de taille 9×9 découpée en 9 carrés de taille 3×3, numéroté de 1 à 9 en partant du carré en haut à gauche et en finissant au carré en bas à droite.

Le but du jeu est de remplir le tableau de chiffres allant de 1 à 9, de sorte à ce que :
- ▷ chaque ligne et chaque colonne ne contiennent qu'une seule fois chaque entier de 1 à 9.
- ▷ chacun des carrés de taille 3×3 ne contiennent qu'une seule fois chaque entier de 1 à 9.

On dira alors que la grille est **bien remplie**.

8	1	3	9	2	5	7	4	6
9	5	6	8	4	7	3	1	2
4	7	2	3	6	1	8	9	5
6	2	4	7	1	9	5	3	8
7	9	5	6	3	8	4	2	1
3	8	1	4	5	2	9	6	7
2	3	8	1	7	4	6	5	9
5	4	9	2	8	6	1	7	3
1	6	7	5	9	3	2	8	4

On représente en Python une grille de Sudoku par un tableau à deux dimensions du module *numpy*.

Vérifier si un grille est correcte

Dans toute cette partie on étudie une grille entièrement remplie. On se demande si cette dernière est **bien** remplie.

1. Écrire une fonction $Recherche(elt, L)$ qui renvoie la liste des positions de l'élément elt dans la liste L. Si l'élément n'est pas trouvé dans la liste, la fonction devra retourner -1.
2. A l'aide de la fonction précédente, écrire une fonction $Ligne\_Bien\_Remplie(S, i)$ qui renvoie $True$ si la ligne i du sudoku S est bien remplie et $False$ sinon.
3. De la même façon écrire une fonction $Colonne\_Bien\_Remplie(S, j)$.
4. (a) Donner, en fonction du nombre k du carré de 3×3, la position (i, j) du nombre situé en haut à gauche de ce dernier.
   Exemple : pour $k = 3$ $i = 3$ et $j = 0$
   (b) Écrire une fonction $Carré\_Bien\_Remplie(S, k)$ qui renvoie $True$ si le $k^{\text{ième}}$ carré du sudoku S est bien remplie et $False$ sinon.
5. Écrire alors une fonction $Sudoku\_Terminé(S)$ qui prend un entrée un sudoku S et renvoie $True$ si le sudoku est bien rempli et $False$ sinon.

Aide à la résolution d'un sudoku

On s'intéresse désormais à une grille partiellement remplie. Pour prendre en compte les cases vides, on les représentera par un zéro dans le tableau du sudoku. On cherche à créer une fonction $chiffresOk$ qui nous aidera à remplir le sudoku en nous indiquant les chiffres qu'on peut mettre dans une case vide de position (i, j) sans enfreindre les règles du jeux.

6. Écrire une fonction $Chiffre\_Ligne(L, i, j)$ qui renvoie la liste des nombres qui apparaissent sur la ligne d'indice i en ne tenant pas compte des zéros ni de l'élément $L[i][j]$.
7. Définir alors de la même manière la fonction $Chiffre\_Colonne(L, i, j)$ qui renvoie la liste des nombres qui apparaissent dans la colonne j en ne tenant pas compte des zéros ni de l'élément $L[i][j]$.

On se donne une case (i, j) avec i et j dans $\{0, ..., 8\}$. On admet que la case en haut à gauche du carré 3×3 auquel appartient la case (i, j) a pour coordonnées :

$$\left(3 \times \left[\frac{i}{3}\right], 3 \times \left[\frac{j}{3}\right]\right)$$

où $[n]$ représente le quotient de la division euclidienne (ou la partie entière du nombre)

8. Écrire une fonction $Chiffre\_Carré(L, i, j)$ qui renvoie la liste des nombres qui apparaissent dans le carré auquel appartient l'élément (i, j) en ne tenant pas compte des zéros ni de l'élément $L[i][j]$.
9. Déduire des questions précédentes une fonction $chiffresOk(L, i, j)$ qui renvoie la liste des chiffres que l'on peut écrire en case (i, j).
10. Écrire une fonction $Solutions\_Dico(S, n)$ qui prend en entrée un sudoku partiellement remplie et renvoie un dictionnaire où :
    ▷ les clés sont des tuples (i, j)
    ▷ les valeurs sont la liste des nombres qu'on peut écrire dans la case (i, j)
    de toutes les cases (i, j) où il est possible d'écrire n nombres.