
Résolution de Sudoku

On s'intéresse dans ce TD au Sudoku. Un Sudoku est une grille carrée avec neuf lignes de neuf cases. Chaque case peut contenir un nombre entre 1 et 9, ou bien être vide. On codera avec 0 les cases vides. On enregistrera dans l'interpréteur Python les grilles sous formes de listes de listes : un Sudoku sera donc la liste de ses 9 lignes, chaque ligne étant constituée par la liste des neuf nombres (0 pour les cases vides) qui y sont inscrits comme dans l'exemple ci-dessous.

Exemple de grille

```
sudoko=[[1,0,0,4,0,0,2,0,0],
        [6,2,0,0,9,8,0,0,7],
        [0,0,0,6,0,0,0,0,5],
        [0,6,2,8,0,9,3,0,0],
        [8,0,7,0,4,0,1,0,9],
        [0,0,1,7,0,3,5,8,0],
        [7,0,0,0,0,6,0,0,0],
        [3,0,0,9,8,0,0,2,1],
        [0,0,9,0,0,7,0,0,3]]
```

Le principe de remplissage du Sudoku est le suivant : dans une ligne donnée doivent apparaître tous les chiffres de 1 à 9, ceux-ci n'apparaissent donc qu'une fois dans chaque ligne. Il doit en être de même dans chaque colonne du Sudoku, et dans chacun des neuf carrés principaux de taille 3×3 qui composent la grande grille.

Exercice 1. *Possibilités dans une case donnée*

Ecrire une fonction, *possibilite(grille, i, j)* qui prend comme arguments le Sudoku *grille*, les numéros *i* et *j* dans $\llbracket 0, 8 \rrbracket$ respectivement de la ligne et de la colonne d'une case donnée, et renvoie comme résultat l'ensemble des chiffres entre 1 et 9 qui ne sont pas encore utilisés dans les autres cases de la ligne *i*, de la colonne *j*, ou dans les autres cases du petit carré 3×3 auquel appartient celle située ligne *i* colonne *j*.

Exercice 2. *Recherche d'une case vide*

Ecrire alors une fonction, *vide(grille)*, qui prend en argument le Sudoku *grille* et qui renvoie un couple $(i, j) \in \llbracket 0, 8 \rrbracket^2$ de coordonnées d'une case vide s'il y en a une dans le sudoku, et qui renvoie *False* sinon.

Exercice 3. *Resolution récursive du problème*

Partant d'un Sudoku *grille*, on va mettre en place un programme *resoudre(grille)* qui en donne une solution ou renvoie *False* s'il n'y en a pas.

L'algorithme est le suivant : on commence par chercher une case vide.

Si l'on n'en trouve pas, c'est résolu : le programme renvoie *grille*.

Si l'on trouve un 0 à la ligne *i* et colonne *j*, on détermine et l'on enregistre les différents chiffres pouvant être inscrits dans la case en question à l'aide de la fonction *possibilite*. L'algorithme va essayer les différentes possibilités dans la case sélectionnée : pour chacune d'entre elles, il faut réutiliser l'algorithme que l'on est en train d'écrire (c'est là l'aspect récursif) avec la grille complétée d'une case afin d'obtenir l'éventuelle solution. Si aucune de ces possibilités ne convient, on renvoie *False*.