

Séance 11: GÉNÉRATION DE GRILLES DE SUDOKU

Université Paris-Diderot

Objectifs:

- Travailler avec des listes de listes
- Utiliser les boucles `for` et `while`

Dans ce TP nous allons proposer une méthode pour manipuler des grilles de Sudoku.

Nous représenterons une grille de sudoku, par une liste de liste de 9 entiers compris entre 0 et 9. Les cases avec 0 sont les cases à deviner, Dans cette liste, chaque sous-liste représente une ligne du sudoku, la première liste représente la première ligne, la deuxième liste la deuxième ligne, etc.

Exercice 1 (À vous de jouer!, ***)

Dans ce qui suit, une **grille de sudoku** fera référence à une liste de neuf listes contenant chacune neuf entiers.

1. Écrire une fonction `grilleSimple` qui ne prend pas d'argument et qui renvoie une liste d'entiers de dimension 2 représentant la grille suivante de sudoku.

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

2. Écrire une fonction `afficheGrille` qui prend en arguments une grille de sudoku et qui l'affiche ligne par ligne sur le terminal en séparant chaque case par un espace.
3. Écrire une fonction `echangeLignes` qui prend en arguments deux numéros de ligne (compris entre 0 et 8) et une grille et échange les deux lignes correspondantes. Cette fonction modifie la grille et ne renvoie rien.

Exemple : Si on échange les lignes 0 et 4 dans la grille dessinée ci-dessus, on obtient la grille suivante.

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

4. Écrire une fonction `echangeColonnes` qui prend en arguments deux numéros de colonnes (compris entre 0 et 8) et une grille et échange les deux colonnes correspondantes. Cette fonction modifie la grille et ne renvoie rien.
5. Nous désirons échanger les lignes de façon aléatoire, mais nous ne voulons échanger que les lignes au sein d'un même bloc, c'est-à-dire que la ligne 0 peut être échangée qu'avec les lignes 1 et 2, la ligne 1 qu'avec les lignes 0 et 2, la ligne 3 qu'avec les lignes 4 et 5, etc. Écrire une fonction `echangeLignesAleatoire` qui prend en argument une grille, choisit aléatoirement quatre entiers i , j et k entre 0 et 2 inclus et échange les lignes $i + 3.j$ et $k + 3.j$.
Indication : Pour tirer des entiers au hasard, utilisez la fonction `randrange` du module `random`, pour cela mettez au début de votre fichier `from random import randrange` et `randrange(a, b)` renvoie un entier au hasard entre a et b , où b est exclu.
6. De la même façon, nous désirons échanger les colonnes de façon aléatoire, mais nous ne voulons échanger que les colonnes au sein d'un même bloc, comme pour les lignes. Écrire une fonction `echangeColonnesAleatoire` qui prend en arguments une grille, choisit aléatoirement trois entiers i , j et k entre 0 et 2 inclus et échange les colonnes $i + 3.j$ et $k + 3.j$.
7. Écrire une fonction `produitGrillePleineAleatoire` qui prend en argument un entier n et commence par créer la grille initiale dessinée au début de ce Tp puis fait les actions suivantes n fois de suite : tire un entier au hasard entre 0 et 1, si cet entier vaut 0, elle échange deux lignes d'un même bloc de façon aléatoire et si cet entier vaut 1, elle échange deux colonnes d'un même bloc de façon aléatoire. La fonction renvoie la grille obtenue. Pensez à tester votre fonction grâce à la fonction d'affichage.
8. Nous disposons maintenant des ingrédients pour produire une grille de sudoku. Écrire une fonction `produit grille` qui prend en arguments deux entiers n et m et qui produit une grille comme à la question précédente en utilisant n comme arguments et qui répète ensuite m fois les actions suivantes : elle tire au hasard deux entiers i et j entre 0 et 8 inclus et elle met un 0 dans la case située à la ligne i et à la colonne j . Le 0 correspond ici à une case à remplir.
On notera que cette fonction ne construit pas une vraie grille de sudoku, étant donné que pour une vraie grille il ne devrait y avoir qu'une seule façon de remplir les cases et notre méthode ne garantit pas ce point.
9. On veut maintenant écrire une fonction `aide` qui étant donné une grille avec des cases à remplir (dans notre cas avec des 0) et les coordonnées d'une case (contenant un 0) donne la liste des valeurs possibles. Une valeur est admissible si elle n'apparaît sur la ligne ou figure la case, ni sur la colonne ni dans le carré. Écrire la fonction `aide` qui prendra donc en arguments une grille et deux entiers i et j compris entre 0 et 8 et renverra la liste des valeurs possibles pour la case située à la ligne 0 et la colonne 8 (si dans cette case il y a une valeur différente de 0, la fonction renverra la liste avec un seul élément correspondant à cette valeur).

□