

Temps Limit : 1h 30m

Enregistrer votre script sous le nom “<nom>.examen2.py”. Où <nom> est votre nom.

Un automate cellulaire consiste en un système discrète en 1D qui peut évoluer au cours du temps. On peut représenter l’univers de ce système par un matrice et une règle d’évolution. Chaque ligne de la matrice correspond à une configuration dans le temps, avec l’instant zéro représentant par la ligne zéro.

1. Définir une matrice, A , $N \times N$ de zéros où la ligne 0 a une valeur de 1 à la position $N/2$ (avec $N = 256$).
2. Dans le cas le plus simple, il ya 256 règles d’évolution possibles. La règle 90 est :

$$A_{i,j} = \begin{cases} 0 & \text{si } A_{i-1,j-1} \neq A_{i-1,j+1} \\ 1 & \text{autrement} \end{cases} \quad (1)$$

Écrire une fonction qui calcule chaque valeur de A à base de cette règle et de mettre à jour la valeur de A .

3. Utiliser la fonction :

plt.imshow(A)

pour voir la matrice résultante et enregistrer le figure dans le fichier **sierpinski.png**. Cette figure s’appelle “triangle de Sierpiński” et il s’agit d’une object fractale avec dimension 1.585.

4. Si nous multiplions A par un vecteur de 1, nous pouvons facilement compter le nombre de entrées non zéro de la matrice A en fonction du temps (ou de la ligne). Tracer le nombre d’entrées non nulles en fonction du temps dans le fichier **sierpinski_zero.png**. On appelle cette séquence de séquence de Gould et correspond à 2^k , où k est le nombre de zéros dans la représentation binaire de temps.
5. Écrire une fonction qui calcule les valeurs d’une matrice B en utilisant la règle 110 :

$$B_{i,j} = \begin{cases} 1 & \text{si } x = 2 \\ 1 & \text{si } x = 1 \text{ et } B_{i-1,j-1} = 0 \\ 0 & \text{autrement} \end{cases} \quad (2)$$

où :

$$x = B_{i-1,j-1} + B_{i-1,j} + B_{i-1,j+1}. \quad (3)$$

Cette règle est Turing complet : quelque programme de calcul ou d’un ordinateur peut être simulé en utilisant cet automate!

6. Calculer la valeur de B à partir d’une matrice de zéros avec une valeur de 1 à la position $N/2$ de la ligne 0.
7. Enregistrer le figure de la matrice B dans le fichier **rule110.png**.