TP 12 informatique

BCPST 1 2019-2020

Matrices

Exercices

Exercice 1. Écrire une fonction `hein(n,p)` qui prend en arguments deux entiers n et p et qui retourne la matrice de taille $n \times p$ contenant que des un.

Exercice 2. Écrire une fonction `Identite(n)` qui prend en arguments un entier positif et qui retourne la matrice I_n.

Exercice 3. Écrire une fonction `Calcul(n,p)` qui prend en arguments un entier positif et qui retourne la matrice de taille $n \times p$ dont le coefficient en position i, j est $i^2 - j + 7$.

Exercice 4. Écrire une fonction qui prend en arguments deux matrices et qui retourne la somme des deux.

Exercice 5. Écrire une fonction qui prend en arguments deux matrices et qui retourne leur produit.

Exercice 6. Il est possible d'effectuer des tirages aléatoires à l'aide de la bibliothèque `random`. Pour cela, on saisit `import random`. La commande random.randint(0,n) permet de tirer un entier au hasard de l'ensemble $\{0, 1, \cdots, n\}$. Écrire une fonction `aleatoire(n,p)` qui retourne une matrice de taille $n \times p$ dont les coefficients ont été tiré au hasard entre 0 et 2.

Exercice 7. $\star$ On considère une population constituée de 1 et de 2 répartie dans une matrice A de taille $n \times p$. Dans notre cas, une case "vide" sera représentée par un 0.

Chacun des nombres $i \in \{1, 2\}$ obéit au comportement suivant : si i a plus de voisins qui lui ressemblent, il est heureux et ne déménage pas. Par contre, si i a strictement plus de voisins différents de lui, il ne se sent pas à sa place et choisit de déménager vers des horizons différents de manière aléatoire.

Par exemple, dans la configuration

$$\begin{pmatrix} 1 & \boxed{2} & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 \end{pmatrix}$$

2 se sent isoler et choisit de déménager de manière aléatoire. On obtient alors par exemple la configuration

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 2 \\ 0 & 1 & 2 & 0 \end{pmatrix}$$

1. Écrire une fonction `voisins(M,i,j)` qui prend en argument une matrice M, et entiers i, j et qui renvoie une liste [pareil,diff], "pareil" étant le nombre de voisins identiques et "diff" le nombre de voisins différents.
2. Écrire une fonction `demenagement(M,i,j)` qui effectue le déménagement de l'élément en position i, j si celui-ci doit avoir lieu.
3. Écrire une fonction `Passe(M)` qui passe en revue toutes les cases de la matrice et qui effectue un déménagement si l'élément visité doit effectivement déménager.
4. Écrire une fonction `evolution(M,n)` qui effectue n passes à la matrice M.
5. À l'aide de la fonction `aleatoire`, on pourra effectuer quelques simulations. Pour un affichage plus visuel, on pourra utiliser la fonction suivante :

```
def affichage(M) :
    import matplotlib.pyplot as plt
    plt.imshow(M)
```

Ainsi, en saisissant `affichage(M)` la matrice sera représenté par une image à trois couleurs, les 0 étant représenté par le bleu, les 1 par le vert et le 2 par le rouge.