TP 08

Matrice de pixels et images

Savoirs et compétences :

- ❒ *AN.C3 : S'interroger sur l'efficacité algorithmique temporelle d'un algorithme*
- ❒ *AN.S1 : Recherche dans une liste, recherche du maximum dans une liste de nombres, calcul de la moyenne et de la variance.*
- ❒ *AN.S2 : Recherche par dichotomie.*
- ❒ *AN.S4 : Recherche d'un mot dans une chaîne de caractères.*

Proposition de corrigé

Activité 1 – Traitement d'images

Exercice 1 –

Question 1

```
def sauve_image (img, N, nom_de_fichier) :
    """f doit être entré sous forme de chaÃ@ne"""
    n = len(img) # n : hauteur = nombre de lignes de la matrice
    p = len(img[0]) # p: largeur =nombre de colonnes de la matrice
    with open(nom_de_fichier,'w') as f :
        f.write("P2\mathbb{N}"+str(p)+" "+str(n)+" "+str(N))
        # Ligne 1 : P2
        # Ligne 2 : n p N
        for i in range(n) : # on écrit les lignes les unes après les autres
            for j in range(p) : # on écrit les éléments de la i ème ligne
                f.write("\mathbb{N}"+str(img[i][j]))
    return None
```

Question 2

```
def sauve_rectangle_noir (n, p, N, nom_de_fichier) :
    """Sauve un rectangle noir nxp, intensité N, dans nom_de_fichier"""
    sauve_image([[0]*p]*n, N, nom_de_fichier)
```

Question 3

```
def sauve_rectangle_blanc (n, p, N, nom_de_fichier) :
    """Sauve un rectangle noir nxp, intensité N, dans nom_de_fichier"""
    sauve_image([[N]*p]*n, N, nom_de_fichier)
```

Question 4

```
def matrice_echiquier (p, N) :
    """Crée la matrice d'un échiquier dont chaque case
    a p pixels de large, et où le blanc correspond à
    l'entier N """
    img = [] # sera la matrice de notre échiquier
    ligne_blanche = [N]*p #ligne du haut d'une case blanche
    ligne_noire = [0]*p #ligne du haut d'une case blanche
    ligne_rangee_paire = (ligne_blanche+ligne_noire)*4
    #ligne de pixels de la matrice, commenÃ§ant par une case blanche
```

```
    ligne_rangee_impaire = (ligne_noire+ligne_blanche)*4
    #ligne de pixels de la matrice, commenÃ§ant par une case blanche
    for i in range(4) : # il y a quatre blocs de la forme
        # "rangée paire, rangée impaire" dans un échiquier
        for j in range (p) : # on ajoute une rangée paire
            img.append(ligne_rangee_paire)
        for j in range (p) : # on ajoute une rangée paire
            img.append(ligne_rangee_impaire)
    return img


def sauve_echiquier (p, N, nom_de_fichier) :
    sauve_image (matrice_echiquier (p, N), N, nom_de_fichier)
```

Question 5

```
def lit_valeurs(nom_de_fichier):
    """Lit le contenu du fichier image f et retourne la liste des
      valeurs lues (séparées par des blancs) sous forme d'une liste
      de chaines de caractères. La première valeur est normalement
      'P2'."""
    with open(nom_de_fichier, 'r') as f:
        c = f.read()
    return c.split()

def lit_image(nom_de_fichier) :
    """ Lit le contenu du fichier image f au format PGM et renvoie
    la matrice et la valeur N du blanc correspondantes """
    L = lit_valeurs (nom_de_fichier)
    # Liste des valeurs de f, l'élément d'indice 0 est 'P2',
    n = int(L[2]) # Celui d'indice 2 est n
    p = int(L[1]) # Celui d'indice 1 est p
    N = int(L[3]) # Celui d'indice 3 est N
    img = [[0]*p for i in range(n)] # créé une matrice vide de taille n x p
    # Attention aux alias !
    for i in range(n) :
        for j in range(p) :
            img[i][j] = int(L[i*p + j + 4])
            # Les valeurs àrentrer dans la matrice ne commencent qu'à l'indice 4 ;
            # les valeurs de la ième ligne commencent àl'indice i*p + 4 ;
            # la valeur de l'emplacement (i,j) se trouve àl'indice i*p + 4 + j
    return(img,N)
```

Question 6

```
def negatif (fichier_entree,fichier_sortie) :
    """Renvoie l'image négative de fichier_entree, dans fichier_sortie """
    (img,N) = lit_image(fichier_entree)
    for i in range(len(img)) :
        for j in range(len(img[0])) :
            img[i][j] = N - img[i][j] # 0 est le noir, N le blanc,
            # donc N - a est la couleur symetrique de a
    sauve_image(img,N,fichier_sortie)
    return None
```

Question 7

```
def rotation90 (fichier_entree,fichier_sortie) :
    """Pivote l'image fichier_entree de 90 degrés dans le sens trigo
    et enregistre le résultat dans fichier_sortie """
    (img,N) = lit_image(fichier_entree)
    n = len(img)
    p = len(img[0])
    img_sortie = [[img[i][p-j-1] for i in range(n)] for j in range(p)]
```

```
    # on construit une matrice de taille p x n ; regarder sur un
    # exemple simple pour se persuader que les indices sont les bons !
    sauve_image(img_sortie,N,fichier_sortie)
    return None
```