

Exercice 1.

Dans cet exercice, on cherche à écrire une fonction qui effectue la rotation d'une image de 90 degrés en utilisant le principe «diviser pour régner».

Pour éviter les complications inutiles, on considère que l'image a autant de pixels en largeur qu'en hauteur et de plus, on considère que cette dimension commune est une puissance de 2, par exemple 512×512 .

L'idée de l'algorithme est de découper l'image en quatre et d'effectuer une rotation circulaire de chacun des quadrants obtenus. On recommence ensuite ce procédé avec chacune des quatre parties.

L'algorithme s'arrête lorsque l'image ne contient plus qu'un seul pixel, la rotation d'une telle image revient à ne rien faire.



Pour manipuler une image avec Python, on peut par exemple utiliser la bibliothèque PIL (Python Image Library) et plus précisément son module `Image`.

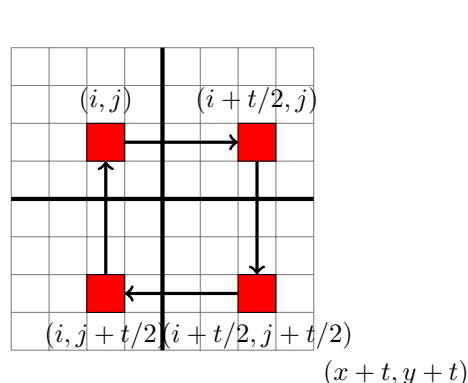
```
from PIL import Image
# on charge l'image contenue dans le fichier chat.png
im = Image.open("chat.png")
# on obtient les dimensions de l'image
largeur, hauteur = im.size
# on obtient la matrice des pixels constituant l'image
px = im.load()
```

Pour $0 \leq x < \text{largeur}$ et $0 \leq y < \text{hauteur}$, la couleur du pixel (x, y) est donnée par $\text{p}[x, y]$.

1. La fonction `echange_circulaire(px, x, y, t)` réalise la permutation des quadrants de la portion carrée comprise entre les pixels (x, y) et $(x+t, y+t)$. Cette fonction effectue la permutation de chaque pixel comme dans la figure ci-dessous.

Compléter le programme ci-dessous.

```
def echange_circulaire(px, x, y, t):
    n = t//2
    for i in range(x, x+n):
        for j in range(y, y+n):
            tmp = px[i, j]
            px[i, j] = ???
            ???
            ???
            ??? = tmp
```



2. Afin de procéder récursivement, on va définir une fonction `rotation_rec(px, x, y, t)` qui effectue la rotation de la portion carrée comprise entre les pixels (x, y) et $(x+t, y+t)$.

```
def rotation_rec(px, x, y, t):  
    if ???:  
        return  
  
    echange_circulaire(px, x, y, t)  
  
    n = t//2  
    rotation_rec(???)  
    rotation_rec(???)  
    rotation_rec(???)  
    rotation_rec(???)
```

3. Ecrire finalement une fonction `rotation(px)` qui effectue la rotation de l'image entière. Une fois la rotation effectuée, on pourra sauvegarder le résultat dans un autre fichier avec la commande `im.save("chat_rotation.png")`.