assignment06

May 8, 2019

1 K-means clustering on x,y location

2 Name : Jinhyuk-Park

3 Student-ID : 20141775

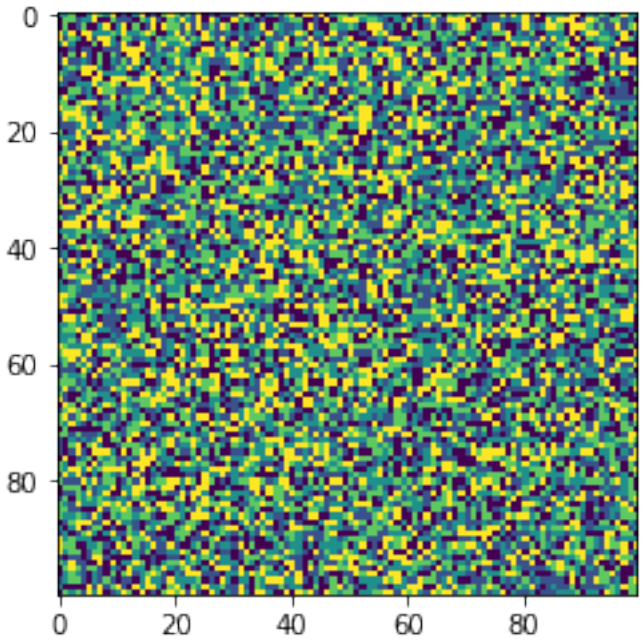
In [36]:
```
import PIL.Image as pilimg
import matplotlib.pyplot as plt
import numpy as np
import random


size_row        = 100
size_col          = 100
x_matrix = np.empty((size_col, size_row), dtype=int)
y_matrix = np.empty((size_col, size_row), dtype=int)

for i in range(0, size_row):
    for j in range(0, size_col):
        x_matrix[i][j] = i
        y_matrix[i][j] = j

#
# normalize the values of the input data to be [0, 1]
#
def normalize(data):

    data_normalized = (data - min(data)) / (max(data) - min(data))

    return(data_normalized)

#
# example of distance function between two vectors x and y
#
def l2_distance(x, y):
```

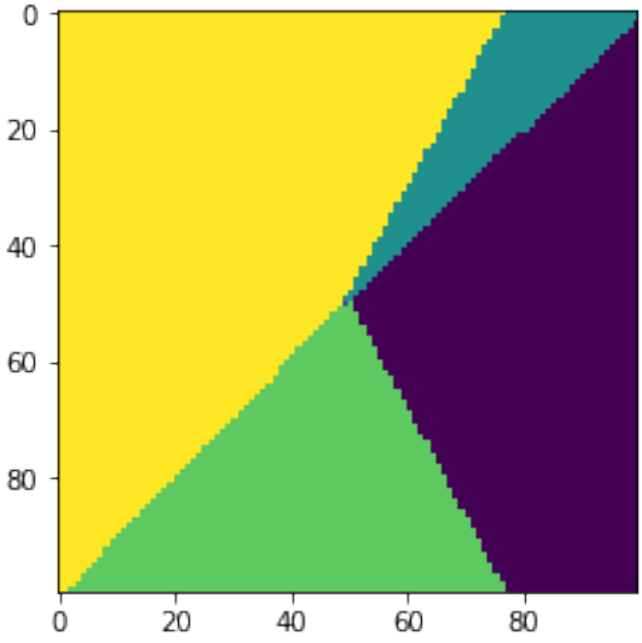
```
    d = (x - y) ** 2
    s = np.sum(d)
    #r = np.sqrt(s)

    return(s)

#
# example of distance function between two vectors x and y
#
def l1_distance(x, y):

    d = abs(x - y)
    s = np.sum(d)

    return(s)

#
# calcuate the values of the input data in l2-norm
#
def norm(x):
    r = np.sqrt(x.T * x)

    return(r)
```

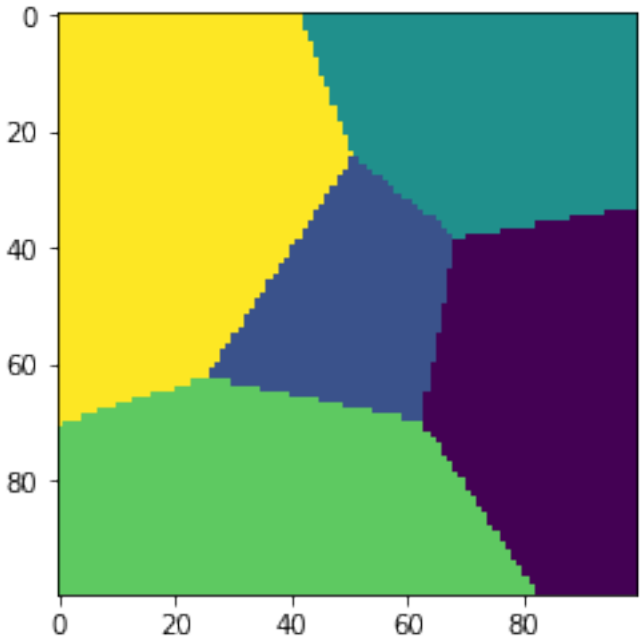
3.1 k = 5 clustering using l2-norm

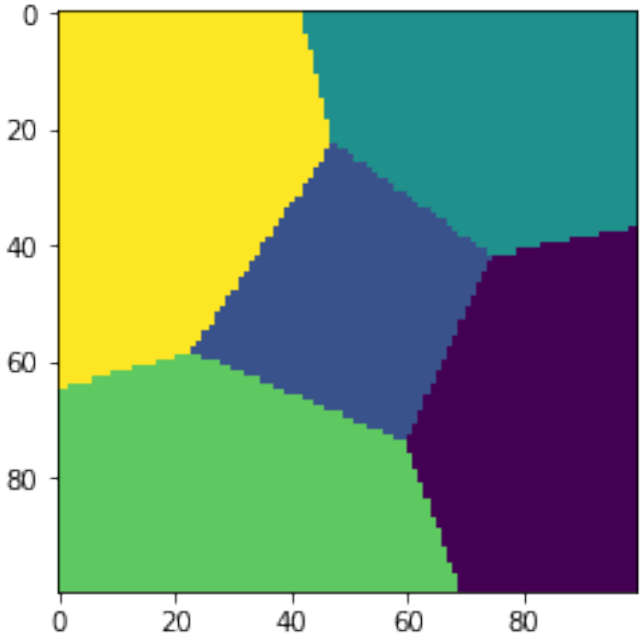
In [37]:
```
k = 5

x_list_centroid = np.zeros(k, dtype=float)
x_list_count    = np.zeros(k)
y_list_centroid = np.zeros(k, dtype=float)
y_list_count    = np.zeros(k)

list_label    = np.empty((size_col, size_row), dtype=int)

for i in range(size_col):
    for j in range(size_row):
        label       = random.randint(0, k - 1)
        list_label[i][j]       = label
        x_list_centroid[label]+= x_matrix[i][j]
        y_list_centroid[label]+= y_matrix[i][j]
        x_list_count[label]      += 1
        y_list_count[label]      += 1


for i in range(0, k):
    x_list_centroid[i] /= x_list_count[i]
    y_list_centroid[i] /= y_list_count[i]
```

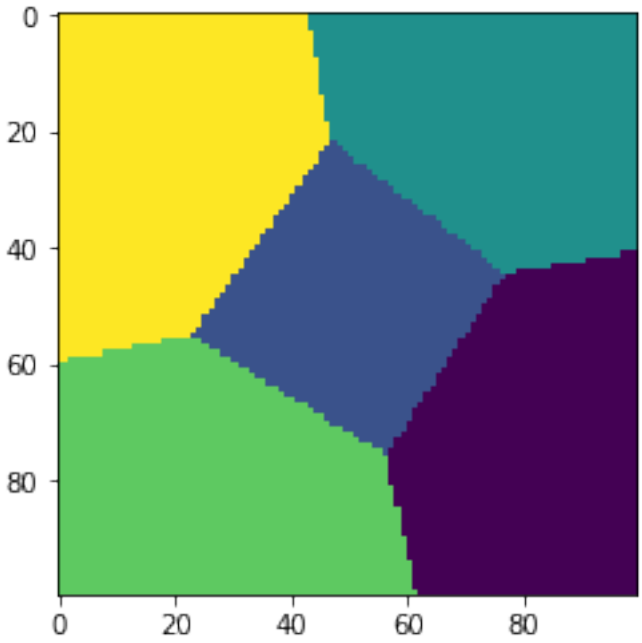
```
while True:
    plt.imshow(list_label)
    plt.show()
    checkUpdate = 0
    for i in range(size_col):
        for j in range(size_row):
            label = list_label[i][j]
            min = l2_distance(x_list_centroid[label], x_matrix[i][j]) + l2_distance(y_

            for m in range(k):
                if m == label:
                    continue

                checkDistance = l2_distance(x_list_centroid[m], x_matrix[i][j]) + l2_c
                if(min > checkDistance):
                    list_label[i][j] = m
                    min = checkDistance
                    checkUpdate += 1

    if(checkUpdate == 0):
        break

    x_list_centroid = np.zeros(k, dtype=float)
    y_list_centroid = np.zeros(k, dtype=float)
    x_list_count = np.zeros(k)
    y_list_count = np.zeros(k)

    for i in range(size_col):
        for j in range(size_row):
            label = list_label[i][j]
            x_list_centroid[label] += x_matrix[i][j]
            y_list_centroid[label] += y_matrix[i][j]
            x_list_count[label]       += 1
            y_list_count[label]       += 1

    for i in range(0, k):
        x_list_centroid[i] /= x_list_count[i]
        y_list_centroid[i] /= y_list_count[i]
```

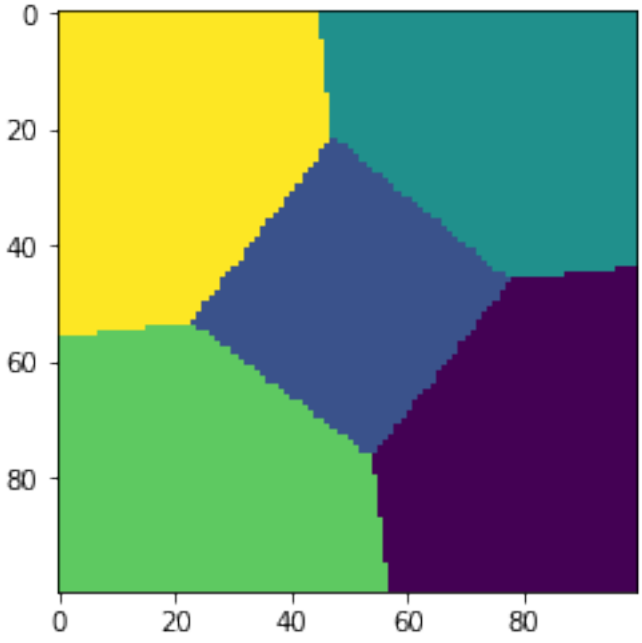

0
20
40
60
80
0
20
40
60
80

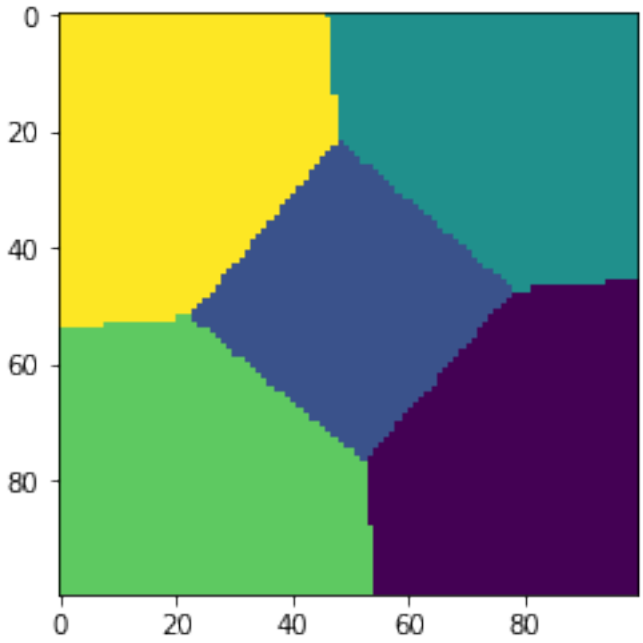

0
20
40
60
80
0
20
40
60
80

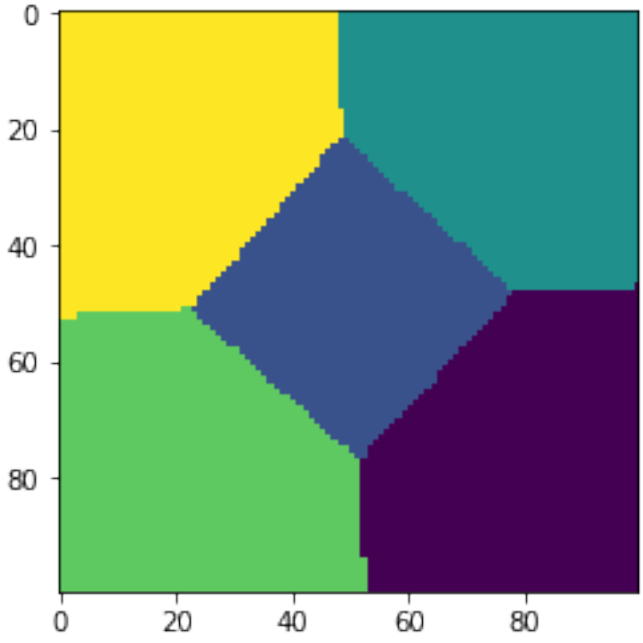

0
20
40
60
80
0
20
40
60
80

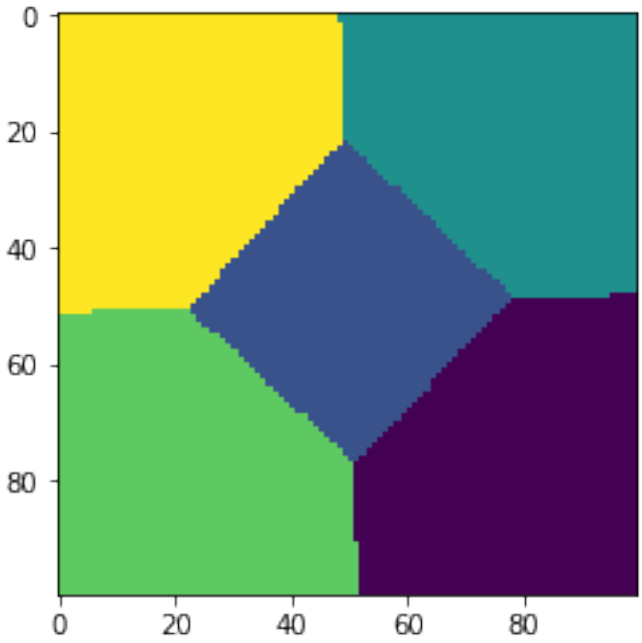

0
20
40
60
80
0
20
40
60
80

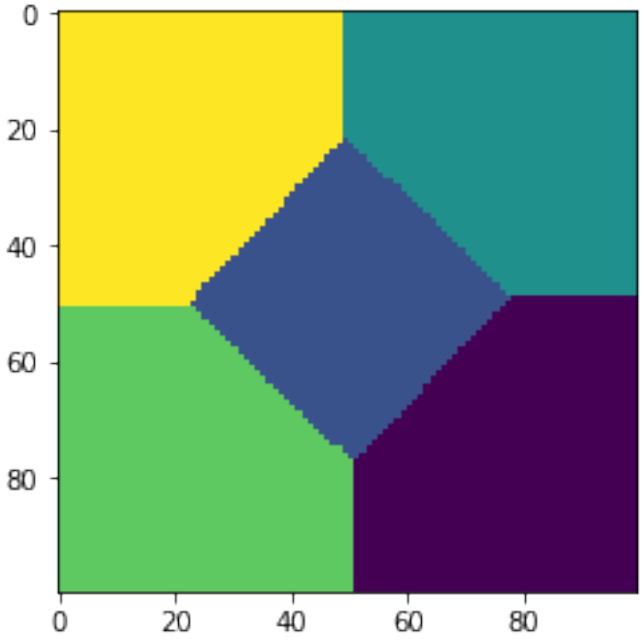

0
20
40
60
80
0
20
40
60
80

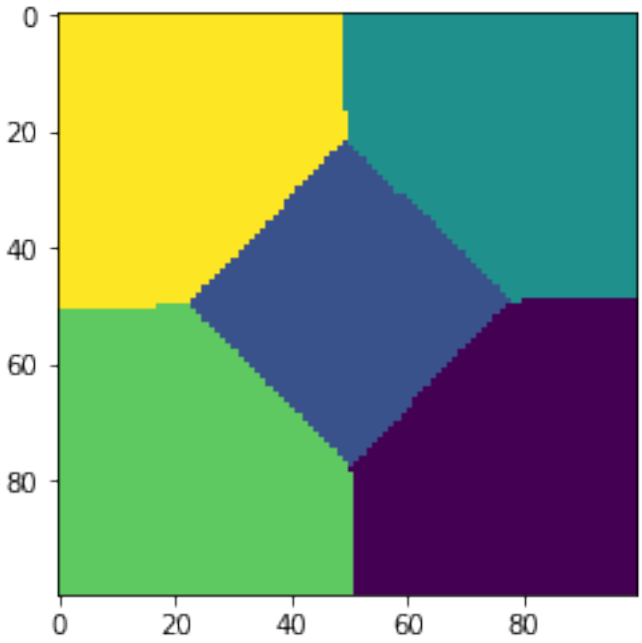

0
20
40
60
80
0
20
40
60
80

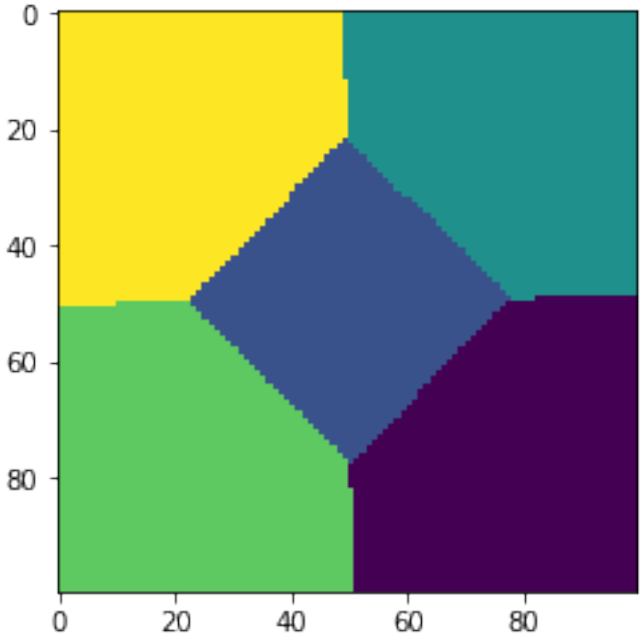

0
20
40
60
80
0
20
40
60
80

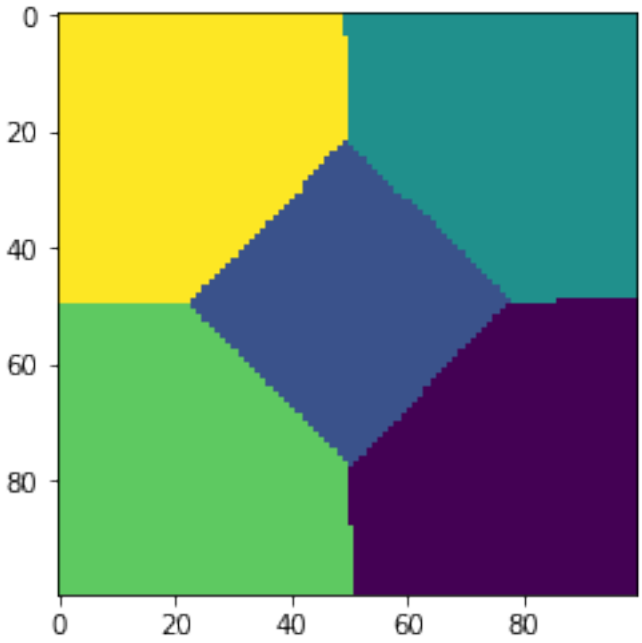

0
20
40
60
80
0
20
40
60
80

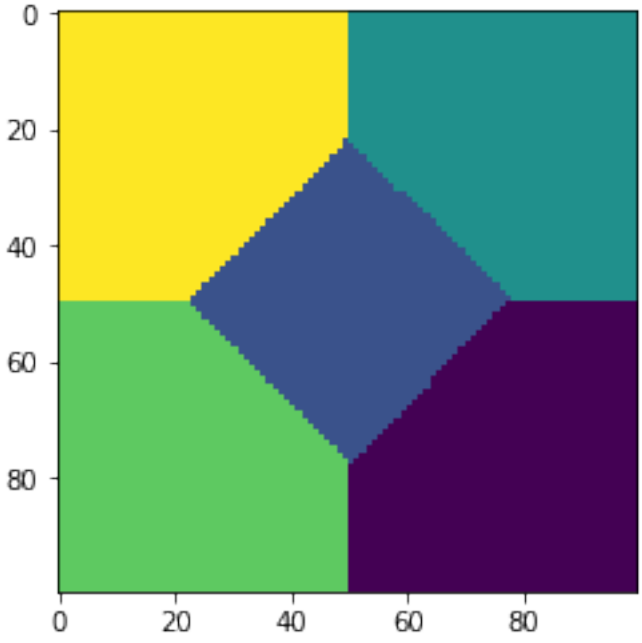

0
20
40
60
80
0
20
40
60
80

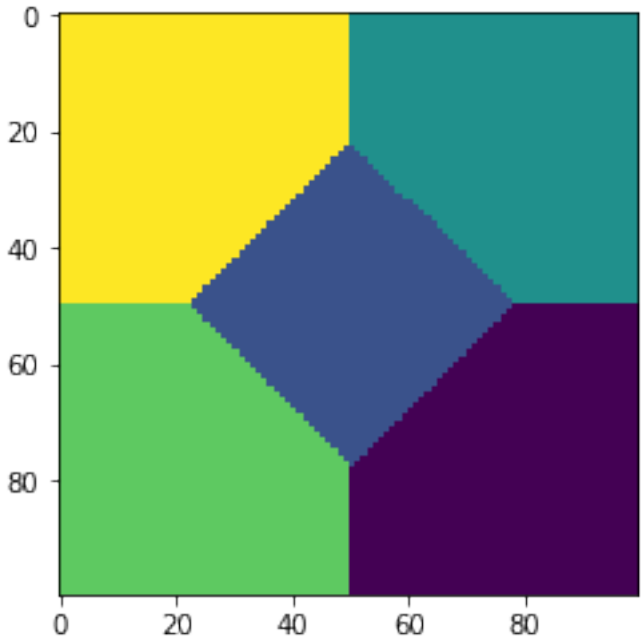

0
20
40
60
80
0
20
40
60
80


0
20
40
60
80
0
20
40
60
80


0
20
40
60
80
0
20
40
60
80


0
20
40
60
80
0
20
40
60
80

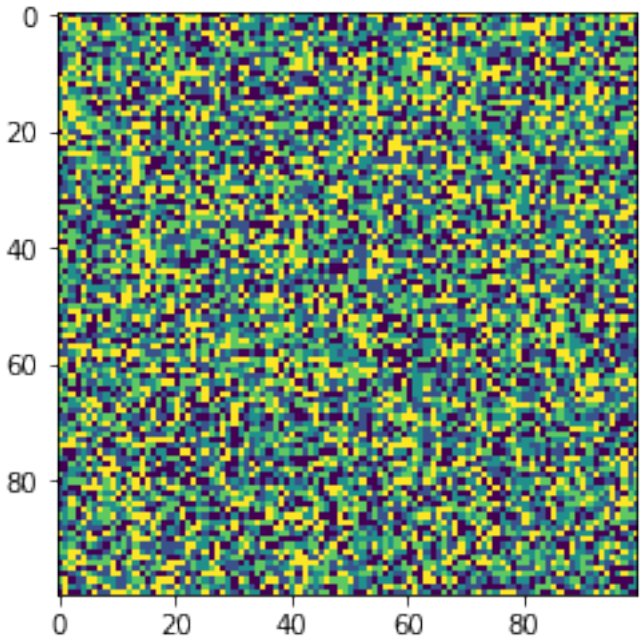

0
20
40
60
80
0
20
40
60
80

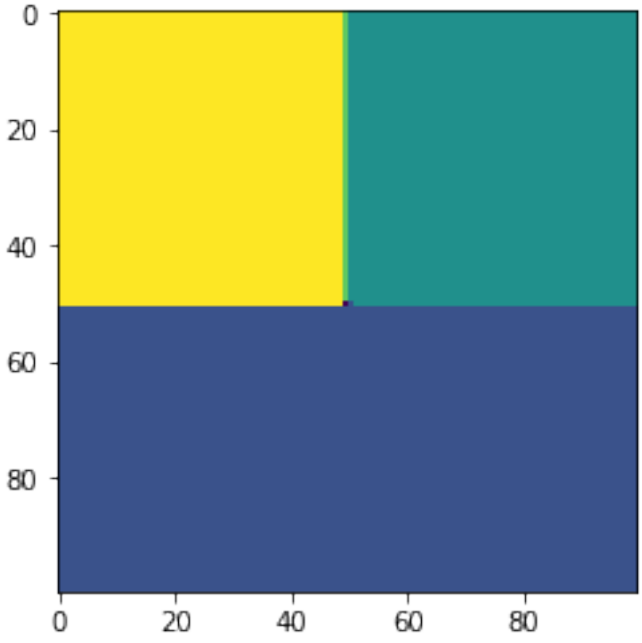


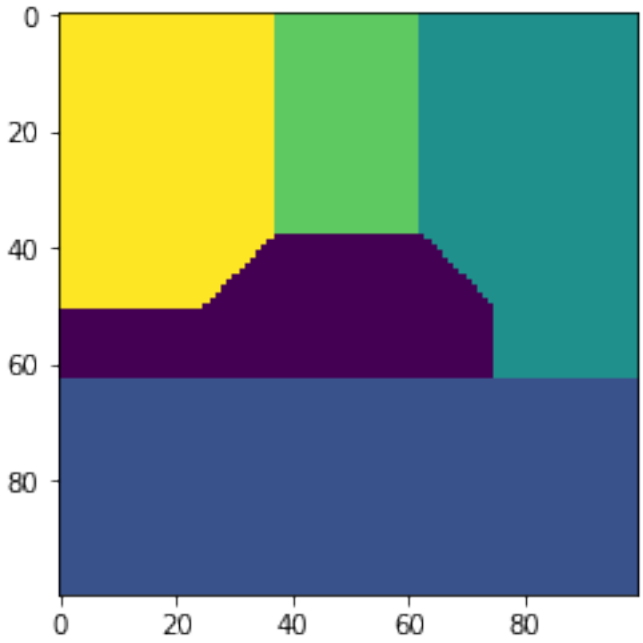
3.2 k = 5 clustering using l1-norm

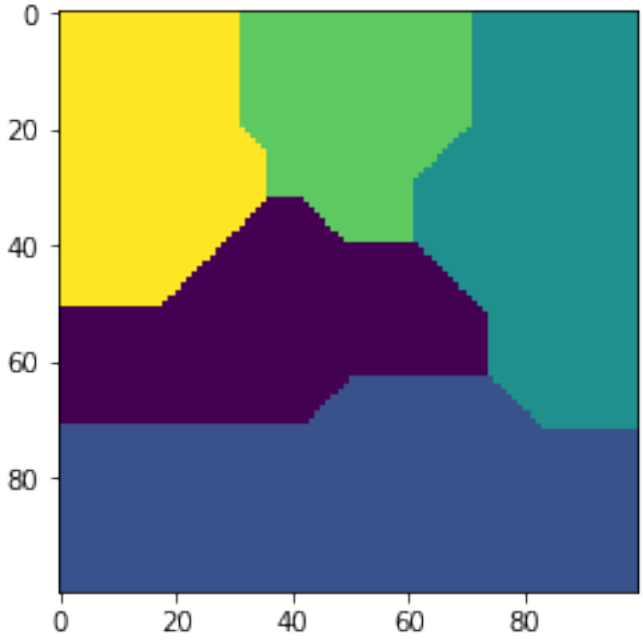
In [38]:
```
k = 5

x_list_centroid = np.zeros(k, dtype=float)
x_list_count    = np.zeros(k)
y_list_centroid = np.zeros(k, dtype=float)
y_list_count    = np.zeros(k)

list_label    = np.empty((size_col, size_row), dtype=int)

for i in range(size_col):
    for j in range(size_row):
        label       = random.randint(0, k - 1)
        list_label[i][j]       = label
        x_list_centroid[label]+= x_matrix[i][j]
        y_list_centroid[label]+= y_matrix[i][j]
        x_list_count[label]      += 1
        y_list_count[label]      += 1


for i in range(0, k):
    x_list_centroid[i] /= x_list_count[i]
    y_list_centroid[i] /= y_list_count[i]
```

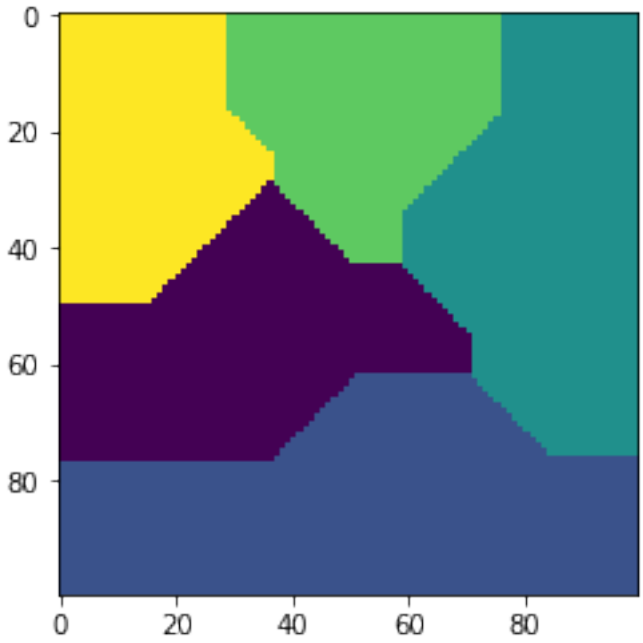
```
while True:
    plt.imshow(list_label)
    plt.show()
    checkUpdate = 0
    for i in range(size_col):
        for j in range(size_row):
            label = list_label[i][j]
            min = l1_distance(x_list_centroid[label], x_matrix[i][j]) + l1_distance(y_

            for m in range(k):
                if m == label:
                    continue

                checkDistance = l1_distance(x_list_centroid[m], x_matrix[i][j]) + l1_d
                if(min > checkDistance):
                    list_label[i][j] = m
                    min = checkDistance
                    checkUpdate += 1

    if(checkUpdate == 0):
        break

    x_list_centroid = np.zeros(k, dtype=float)
    y_list_centroid = np.zeros(k, dtype=float)
    x_list_count = np.zeros(k)
    y_list_count = np.zeros(k)

    for i in range(size_col):
        for j in range(size_row):
            label = list_label[i][j]
            x_list_centroid[label] += x_matrix[i][j]
            y_list_centroid[label] += y_matrix[i][j]
            x_list_count[label]       += 1
            y_list_count[label]       += 1

    for i in range(0, k):
        x_list_centroid[i] /= x_list_count[i]
        y_list_centroid[i] /= y_list_count[i]
```

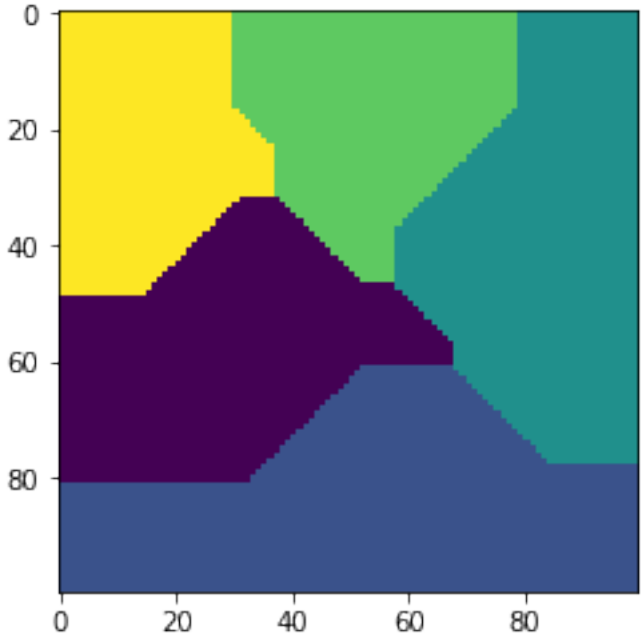

0
20
40
60
80
0
20
40
60
80

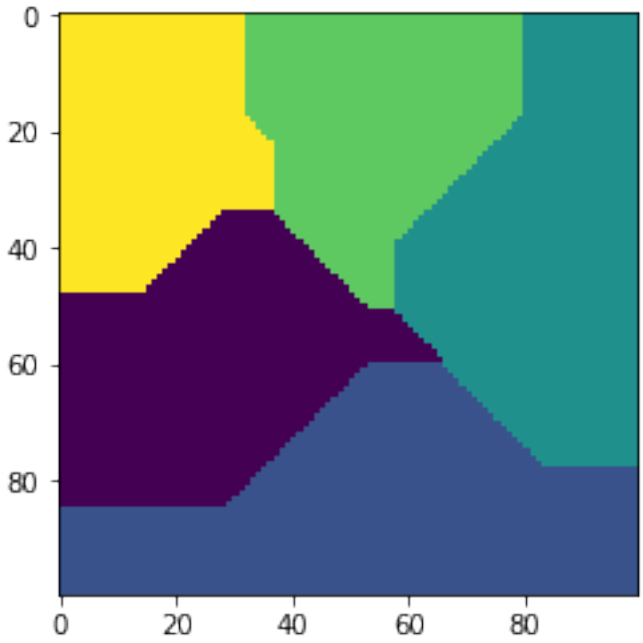

0
20
40
60
80
0
20
40
60
80

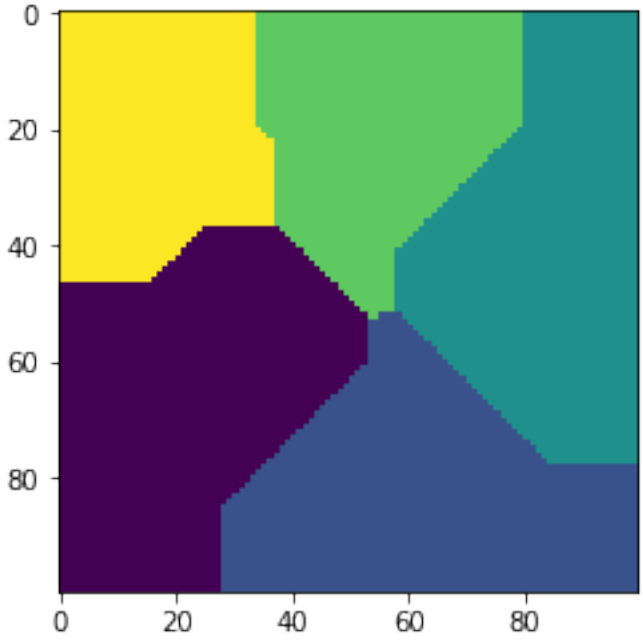

0
20
40
60
80
0
20
40
60
80

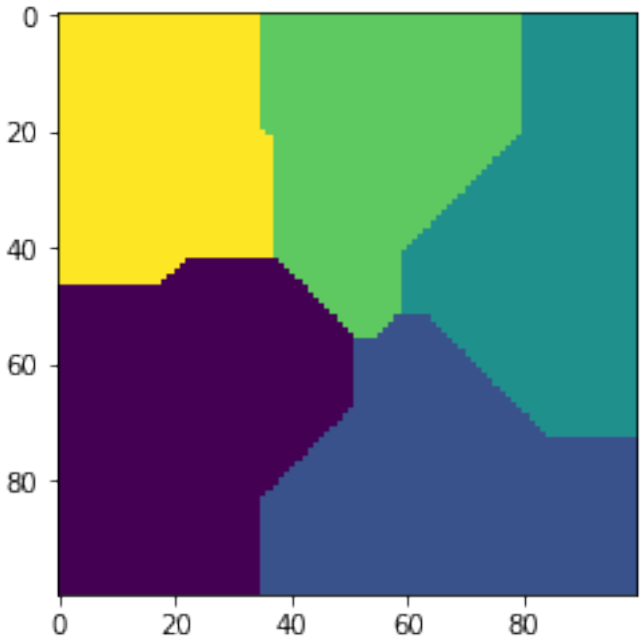

0
20
40
60
80
0
20
40
60
80

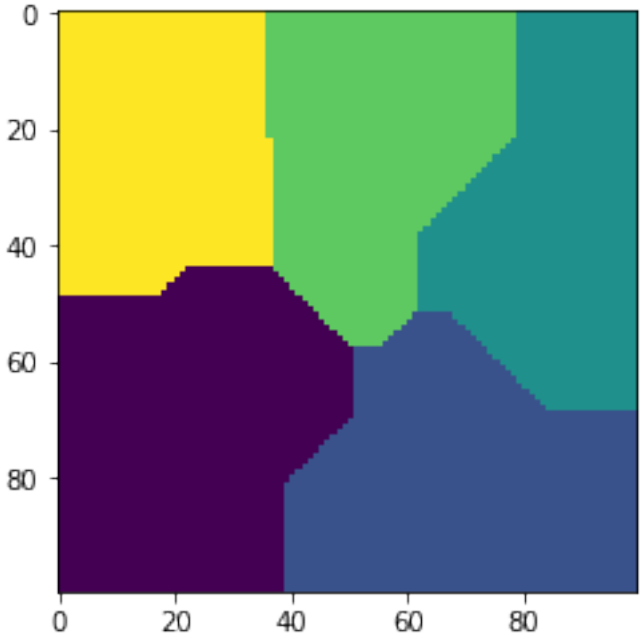

0
20
40
60
80
0
20
40
60
80

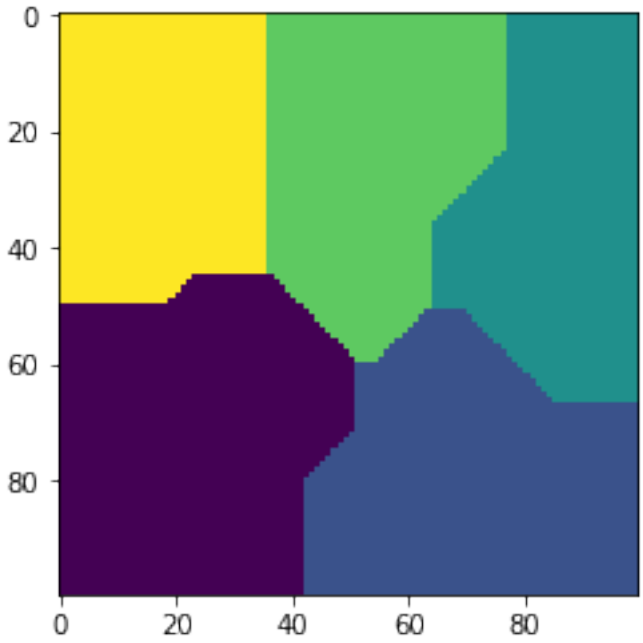

0
20
40
60
80
0
20
40
60
80

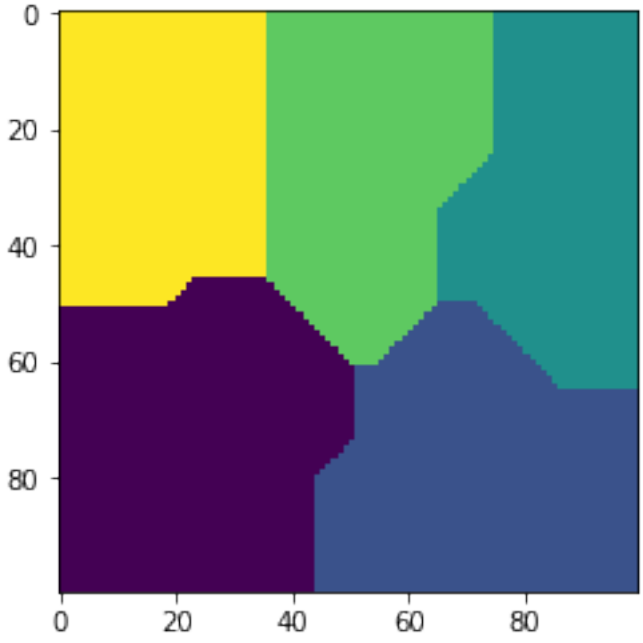

0
20
40
60
80
0
20
40
60
80

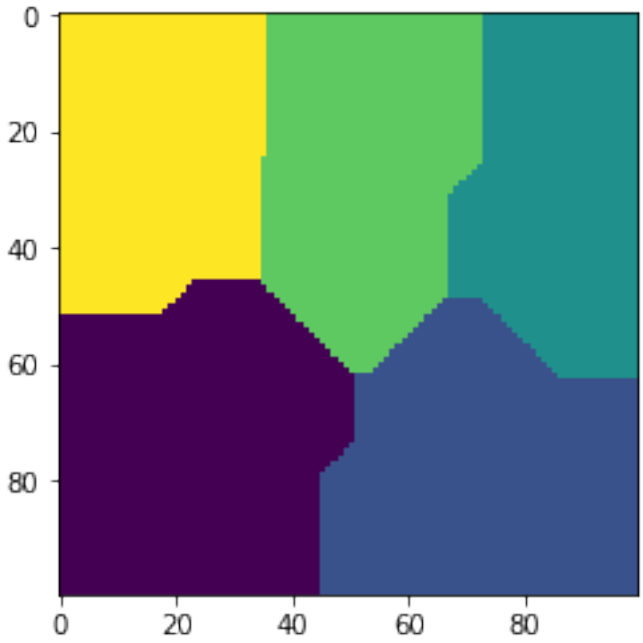

0
20
40
60
80
0
20
40
60
80

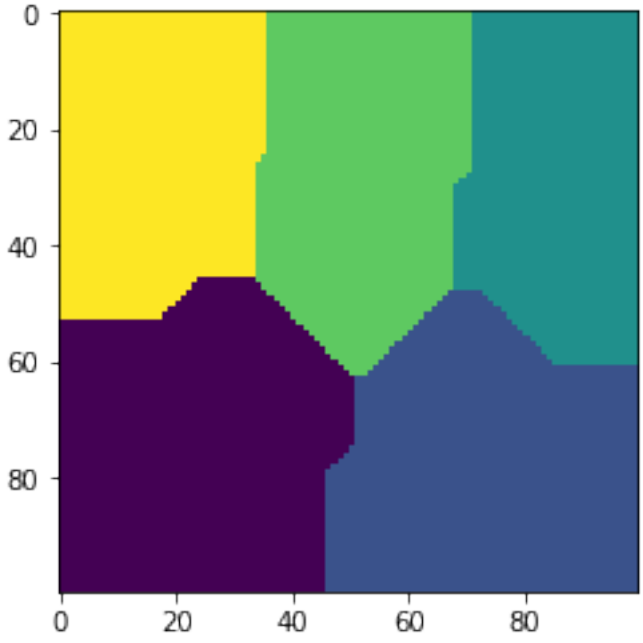

0
20
40
60
80
0
20
40
60
80

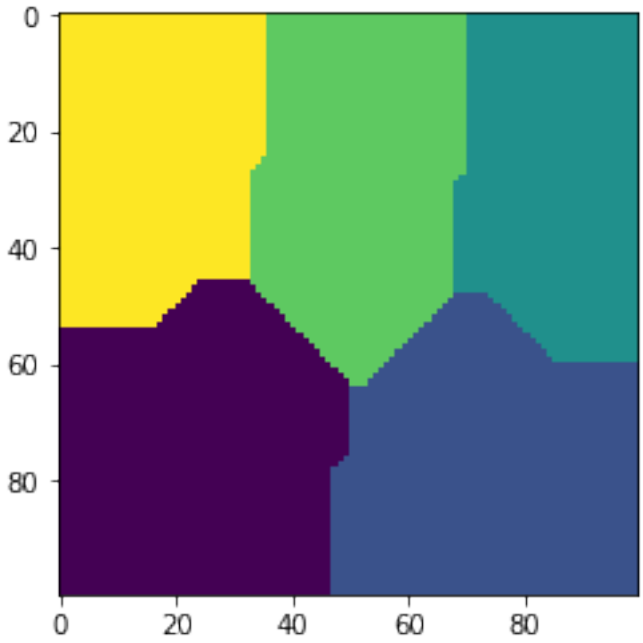

0
20
40
60
80
0
20
40
60
80

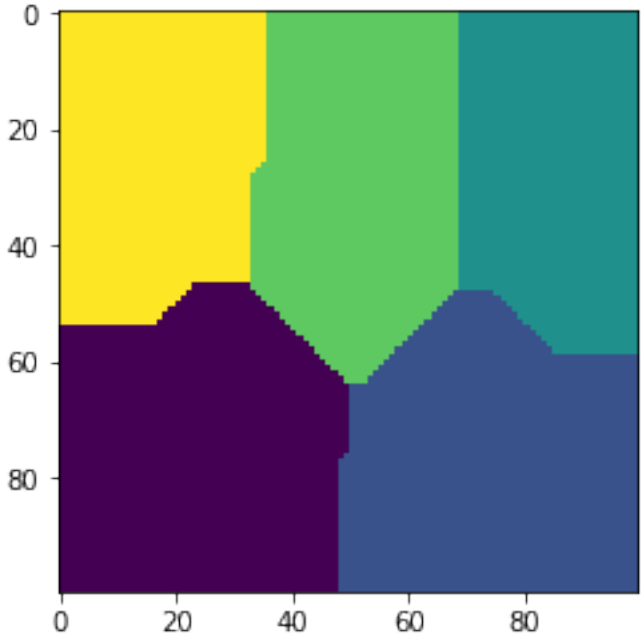

0
20
40
60
80
0
20
40
60
80

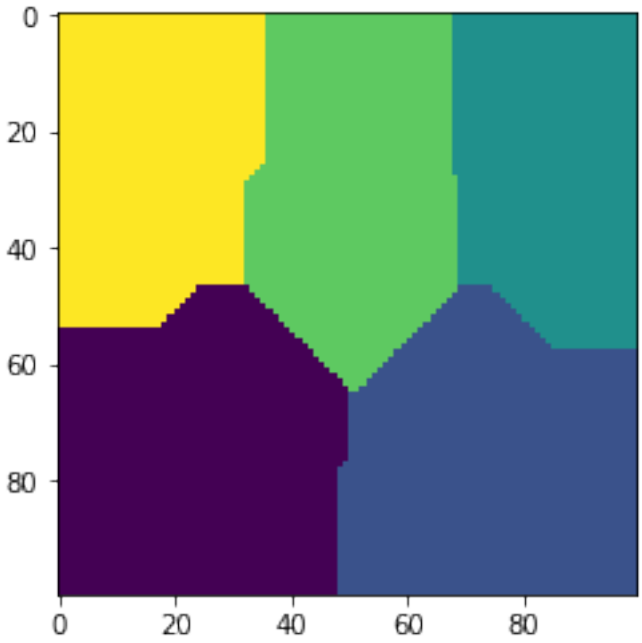

0
20
40
60
80
0
20
40
60
80

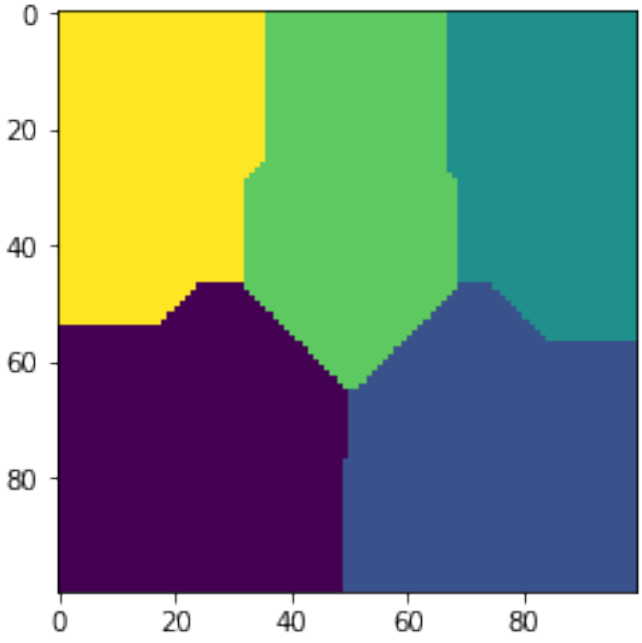

0
20
40
60
80
0
20
40
60
80

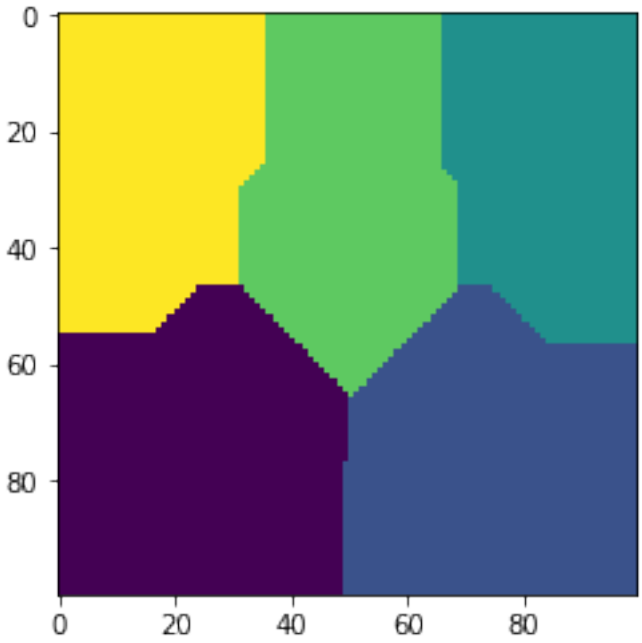

0
20
40
60
80
0
20
40
60
80

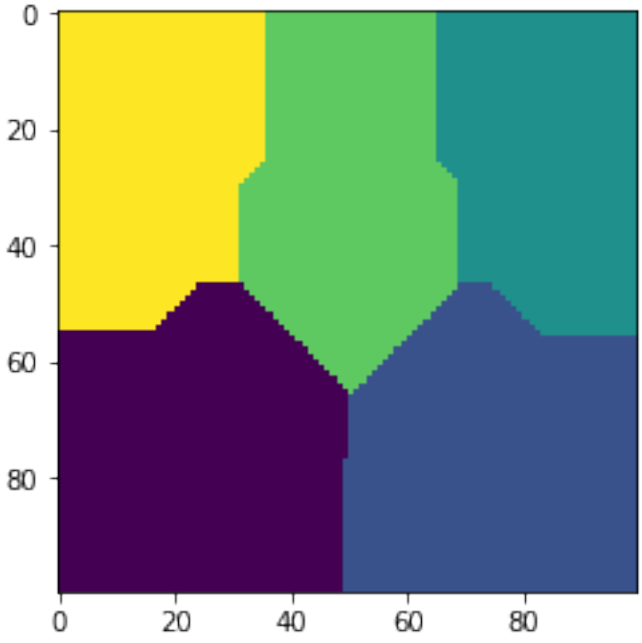

0
20
40
60
80
0
20
40
60
80

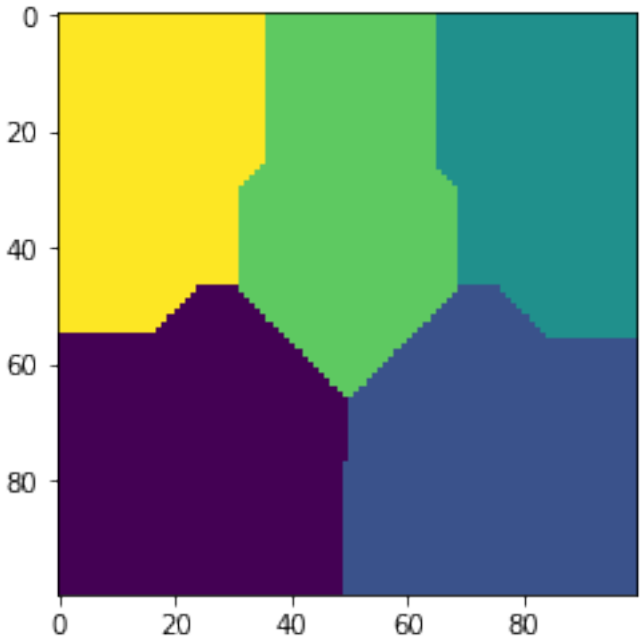

0
20
40
60
80
0
20
40
60
80

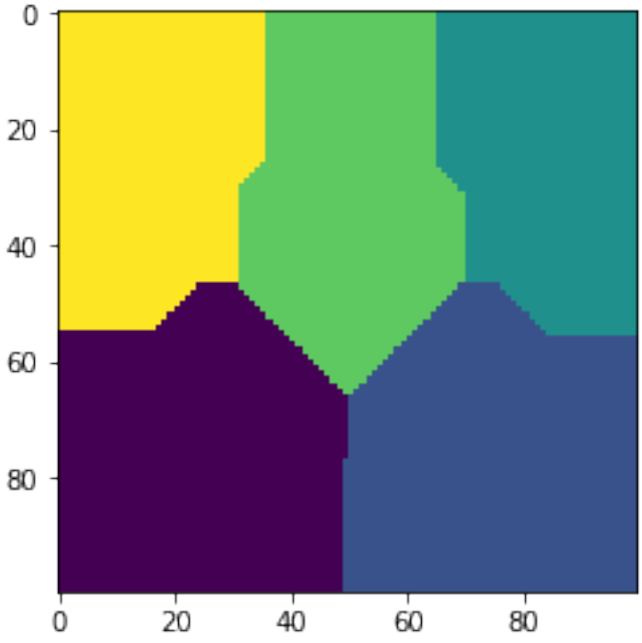

0
20
40
60
80
0
20
40
60
80

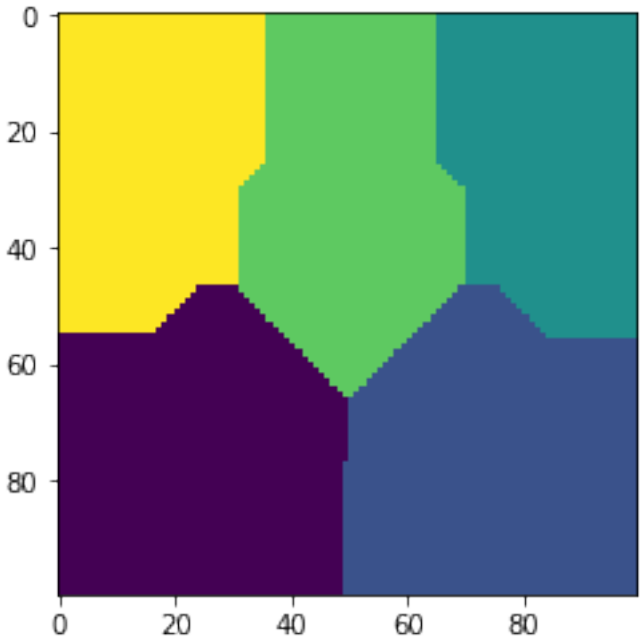

0
20
40
60
80
0
20
40
60
80


0
20
40
60
80
0
20
40
60
80


0
20
40
60
80
0
20
40
60
80


0
20
40
60
80
0
20
40
60
80


0
20
40
60
80
0
20
40
60
80


0
20
40
60
80
0
20
40
60
80