



Εργαστήριο 4 Τεχνητής Όρασης

Κώστας Μαριάς

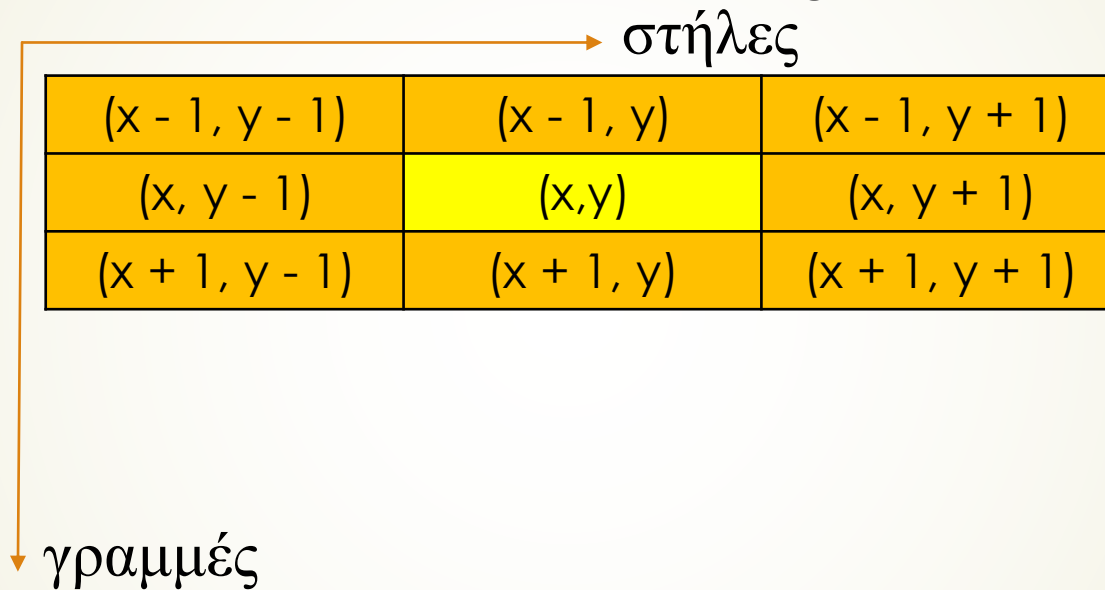
Κώδικες αλυσίδας για την περιγραφή σχημάτων

Σκοπός Εργαστηρίου:

- Να επαναλάβουμε σύντομα τον κώδικα αλυσίδας για την περιγραφή σχήματος.
- Να δούμε αν μπορούμε να τρέξουμε έτοιμες συναρτήσεις για να μας δίνουν τον κώδικα αλυσίδας.
- Να προσπαθήσουμε να φτιάξουμε δικό μας κώδικα.

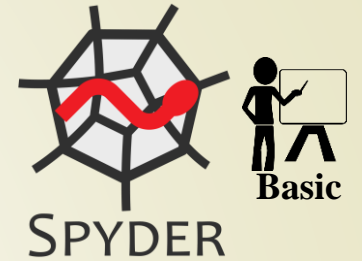
Βασικές σχέσεις ανάμεσα σε pixels

- Γείτονες του pixel p : 8-γείτονες $N_8(p) = N_4(p) + N_D(p)$



- Σε όλες τις περιπτώσεις αν το (x, y) είναι στο περίγραμμα της εικόνας οι γείτονες ενδέχεται να είναι έξω από την εικόνα!!!

Γειτνίαση, Συνδεσιμότητα, Περιοχές , και Όρια

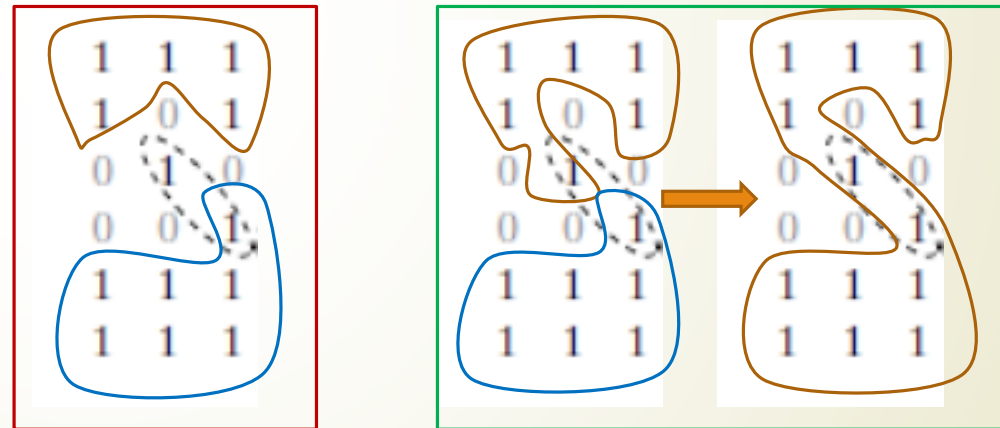


- ▶ Έστω ότι R είναι ένα υποσύνολο των pixels μιας εικόνας.
- ▶ Το R είναι μια περιοχή (*region*) της εικόνας αν είναι ένα συνδεδεμένο σύνολο (*connected set*) δηλ. όλα τα pixels του είναι συνδεδεμένα (υπάρχει ένα μονοπάτι γειτνίασης αναμεσά τους με pixels που ανήκουν αποκλειστικά στο R)
- ▶ Δύο περιοχές είναι γειτονικές αν η ένωσή τους σχηματίζει ένα συνδεδεμένο σύνολο
- ▶ Διαφορετικά οι περιοχές είναι ξεχωριστές (*disjoint*).
- ▶ Για να ορίσουμε όλα τα παραπάνω πρέπει να έχουμε ορίσει πρώτα αν μιλάμε για 4- ή -8, m γειτνίαση περιοχών.

Γειτνίαση, Συνδεσιμότητα, Περιοχές , και Όρια

5

- ▶ Στο παρακάτω παράδειγμα αριστερά οι δύο περιοχές που έχουν **4-γειτνίαση** δεν γειτονεύουν. Αν χρησιμοποιήσουμε όμως **8-γειτνίαση** (δεξιά) οι περιοχές γειτονεύουν γιατί η ένωσή τους είναι ένα συνδεδεμένο σύνολο!



Γειτνίαση, Συνδεσιμότητα, Περιοχές , και Όρια

6

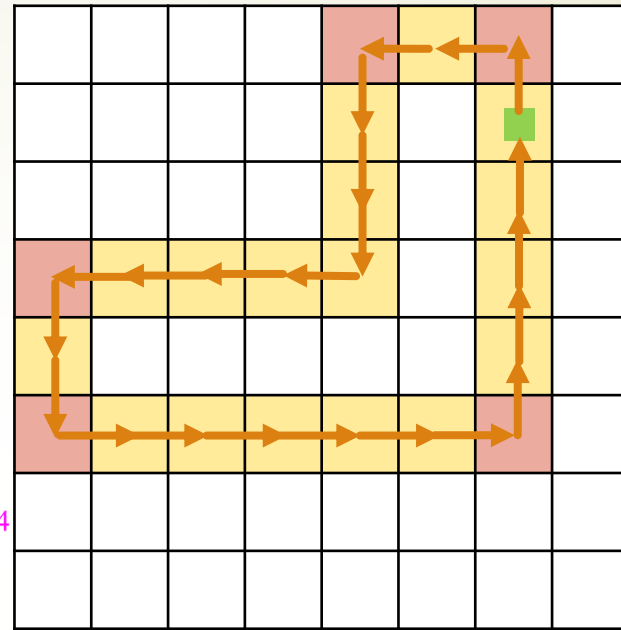
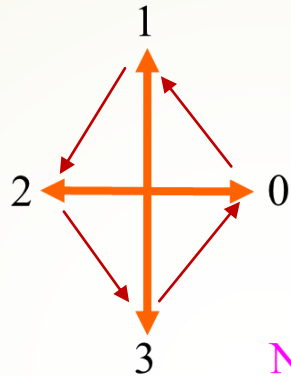
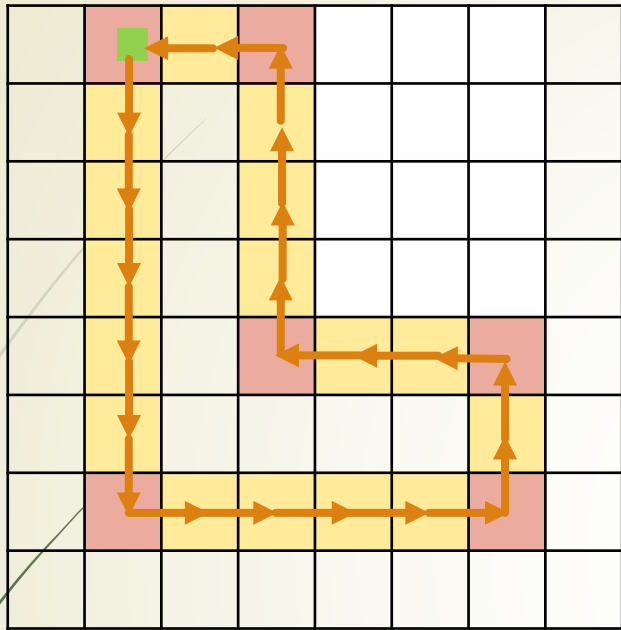
- Ο προηγούμενος ορισμός είναι το εσωτερικό περίγραμμα σε αντίθεση με το εξωτερικό που είναι το αντίστοιχο του background.
- Αυτό είναι σημαντικό για ανάπτυξη αλγορίθμων που εντοπίζουν το περίγραμμα της εικόνας γιατί το περίγραμμα είναι πάντα ένα κλειστό μονοπάτι!

0	0	0
0	1	0
0	1	0
0	1	0
0	1	0
0	0	0

- ✓ Το εσωτερικό περίγραμμα της περιοχής εικόνας με τιμές – 1 είναι η ίδια περιοχή ΑΛΛΑ δεν είναι κλειστό μονοπάτι!!!
- ✓ Το εξωτερικό όμως είναι!!!

Αλυσιδωτοί κώδικες: Μεταβολές λόγω αρχικού σημείου και περιστροφής 90° (1^η Περίπτωση)

7



Αλυσιδωτός κώδικας 1

3 3 3 3 3 3 0 0 0 0 0 1 1 2 2 2 1 1 1 1 2 2
 3 3 3 3 3 3 0 0 0 0 0 1 1 2 2 2 1 1 1 1 2 2 3
0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 3 0 0 0 1 0 1 ✓

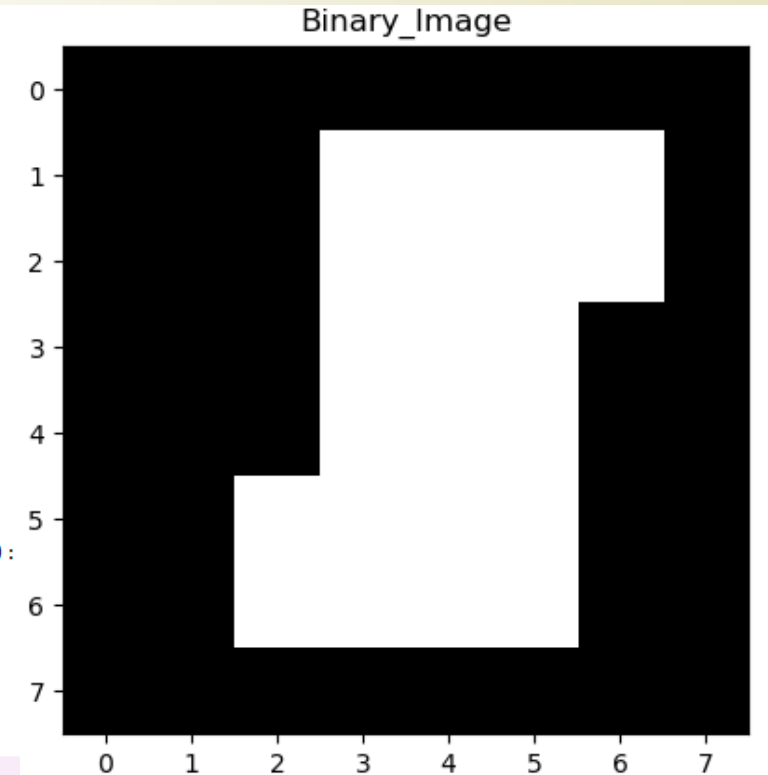
Αλυσιδωτός κώδικας 2

1 2 2 3 3 3 2 2 2 2 3 3 0 0 0 0 0 0 1 1 1 1
 0 0 0 0 0 0 1 1 1 1 1 2 2 3 3 3 2 2 2 2 3 3
 0 0 0 0 0 0 1 1 1 1 1 2 2 3 3 3 2 2 2 2 3 3 0
0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 3 0 0 0 1 0 1 ✓

Οι αριθμοί σχήματος διορθώνουν για την περιστροφή και το σημείο εκκίνησης!

Δημιουργία εικόνας και αποθήκευση συντεταγμένων περιγράμματος

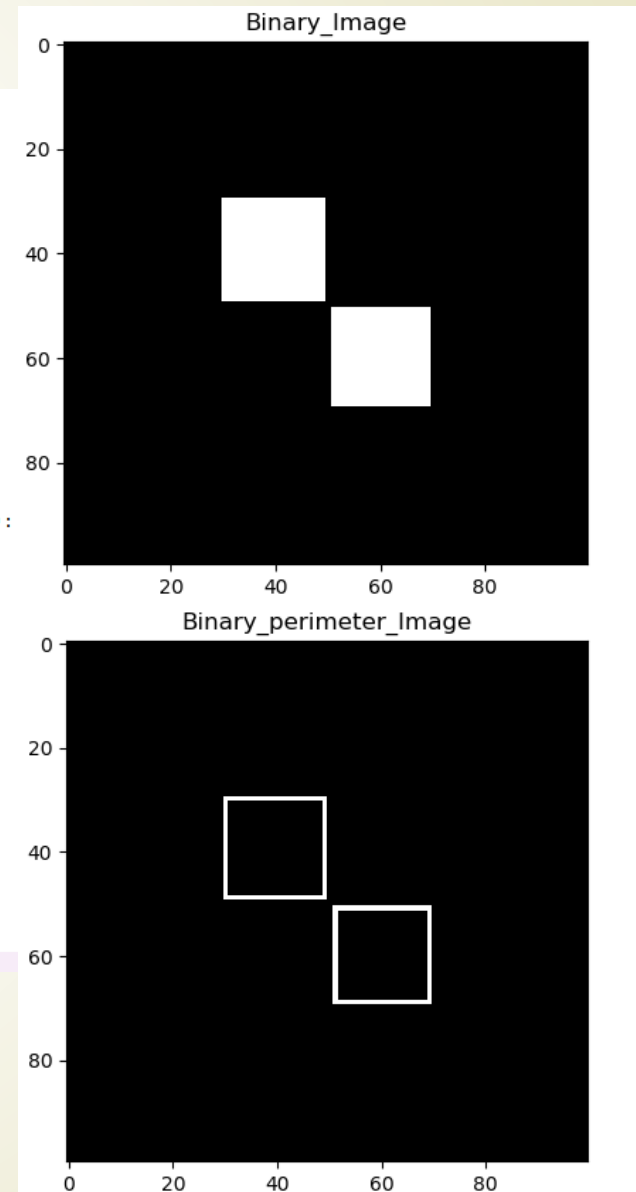
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import skimage as image_tool
4
5 Image_Binary=[[0, 0, 0, 0, 0, 0, 0, 0],
6               [0, 0, 0, 1, 1, 1, 1, 0],
7               [0, 0, 0, 1, 1, 1, 1, 0],
8               [0, 0, 0, 1, 1, 1, 0, 0],
9               [0, 0, 0, 1, 1, 1, 0, 0],
10              [0, 0, 1, 1, 1, 1, 0, 0],
11              [0, 0, 1, 1, 1, 1, 0, 0],
12              [0, 0, 0, 0, 0, 0, 0, 0]]
13
14 Image_Binary=np.asarray(Image_Binary, dtype=np.bool)
15 Image_Binary=Image_Binary.astype(int)
16
17 plt.figure(1)
18 plt.imshow(Image_Binary,cmap=plt.cm.gray)
19 plt.title('Binary_Image')
20
21 cords_x=[]
22 cords_y=[]
23
24 for i in range(len(Image_Binary)-1):
25     for j in range(len(Image_Binary)-1):
26         if (Image_Binary[i,j]==1 and (Image_Binary[i,j-1]==0 or Image_Binary[i,j+1]==0 or Image_Binary[i-1,j]==0 or Image_Binary[i+1,j]==0)):
27             cords_x.append(i)
28             cords_y.append(j)
29
30 cords_x=np.asarray(cords_x, dtype=np.int32)
31 cords_y=np.asarray(cords_y, dtype=np.int32)
32
33 Pinakas_cords=np.transpose(np.vstack((cords_x,cords_y)))
34
35 Overall_connected_list=image_tool.measure.regionprops(Image_Binary)
36 |
```



Δημιουργία εικόνας και αποθήκευση συντεταγμένων/οπτικοποίηση περιγράμματος

9

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import skimage as image_tool
4
5 Image_Binary=np.zeros((100,100))
6
7 Image_Binary[30:50,30:50]=1
8 Image_Binary[51:70,51:70]=1
9
10 Image_Binary=np.asarray(Image_Binary, dtype=np.bool)
11 Image_Binary=Image_Binary.astype(int)
12
13 plt.figure(1)
14 plt.imshow(Image_Binary,cmap=plt.cm.gray)
15 plt.title('Binary_Image')
16
17 cords_x=[]
18 cords_y=[]
19
20 for i in range(len(Image_Binary)-1):
21     for j in range(len(Image_Binary)-1):
22         if (Image_Binary[i,j]==1 and (Image_Binary[i,j-1]==0 or Image_Binary[i,j+1]==0 or Image_Binary[i-1,j]==0 or Image_Binary[i+1,j]==0)):
23             cords_x.append(i)
24             cords_y.append(j)
25
26
27 cords_x=np.asarray(cords_x, dtype=np.int32)
28 cords_y=np.asarray(cords_y, dtype=np.int32)
29
30 Pinakas_cords=np.transpose(np.vstack((cords_x,cords_y)))
31
32 New_Binary=np.zeros((100,100))
33
34 New_Binary=np.asarray(New_Binary, dtype=np.bool)
35 New_Binary=New_Binary.astype(int)
36
37 Overall_connected_list=image_tool.measure.regionprops(Image_Binary)
38
39 New_Binary[Pinakas_cords[:,0],Pinakas_cords[:,1]]=1
40
41 plt.figure(2)
42 plt.imshow(New_Binary,cmap=plt.cm.gray)
43 plt.title('Binary_perimeter_Image')
44
```



Δημιουργία εικόνας και αποθήκευση συντεταγμένων/οπτικοποίηση περιγράμματος 4-8 Γειτνίαση

```

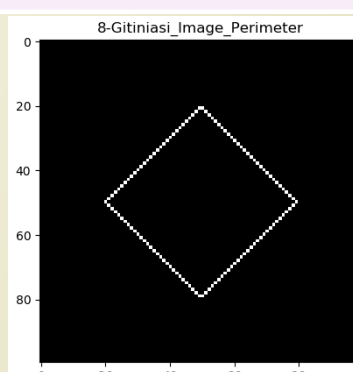
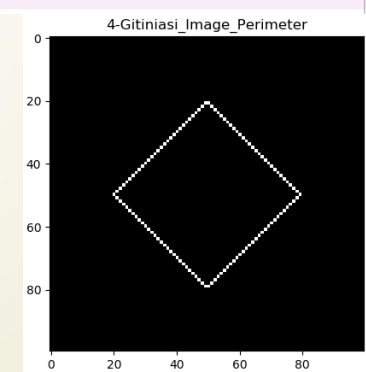
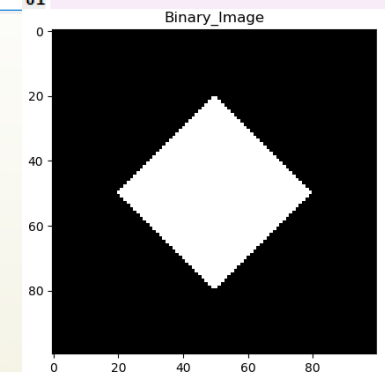
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import skimage as image_tool
4
5 Image_Binary=np.zeros((100,100))
6
7 sum1=0
8 sum2=0
9 for i in range(len(Image_Binary)):
10
11     if(i>= 20 and i<=50):
12         left=50-sum1
13         right=50+sum1
14         Image_Binary[i,left:right]=1
15         sum1=sum1+1
16     if(i>= 50 and i<=80):
17         left=20+sum2
18         right=80-sum2
19         Image_Binary[i,left:right]=1
20         sum2=sum2+1
21
22 Image_Binary=np.asarray(Image_Binary, dtype=np.bool)
23 Image_Binary=Image_Binary.astype(int)
24
25 plt.figure(1)
26 plt.imshow(Image_Binary,cmap=plt.cm.gray)
27 plt.title('Binary_Image')
28
29 cords_x=[]
30 cords_y=[]
31
32 for i in range(len(Image_Binary)-1):
33     for j in range(len(Image_Binary)-1):
34         if (Image_Binary[i,j]==1 and (Image_Binary[i,j-1]==0 or Image_Binary[i,j+1]==0 or Image_Binary[i-1,j]==0 or Image_Binary[i+1,j]==0)):
35             cords_x.append(i)
36             cords_y.append(j)
37
38 cords_x=np.asarray(cords_x, dtype=np.int32)
39 cords_y=np.asarray(cords_y, dtype=np.int32)
40
41 Pinakas_cords=np.transpose(np.vstack((cords_x,cords_y)))
42
43 New_Binary=np.zeros((100,100))
44
45 New_Binary=np.asarray(New_Binary, dtype=np.bool)
46 New_Binary=New_Binary.astype(int)
47
48 Overall_connected_list=image_tool.measure.regionprops(Image_Binary)

```

```

48 Overall_connected_list=image_tool.measure.regionprops(Image_Binary)
49
50 New_Binary[Pinakas_cords[:,0],Pinakas_cords[:,1]]=1
51
52 plt.figure(2)
53 plt.imshow(New_Binary,cmap=plt.cm.gray)
54 plt.title('4-Gitiniasi_Image_Perimeter')
55
56 #-----Coords 8 gitniasi-----
57 new_cords_x2=[]
58 new_cords_y2=[]
59 for i in range(len(New_Binary)-1):
60     for j in range(len(New_Binary)-1):
61         if (New_Binary[i,j]==1 and (New_Binary[i,j-1]==0 or New_Binary[i,j+1]==0 or New_Binary[i-1,j]==0 or New_Binary[i+1,j]==0
62             or New_Binary[i-1,j-1]==0 or New_Binary[i-1,j+1]==0 or New_Binary[i+1,j-1]==0 or New_Binary[i+1,j+1]==0)):
63             new_cords_x2.append(i)
64             new_cords_y2.append(j)
65
66 new_cords_x2=np.asarray(new_cords_x2, dtype=np.int32)
67 new_cords_y2=np.asarray(new_cords_y2, dtype=np.int32)
68
69 Pinakas_cords2=np.transpose(np.vstack((new_cords_x2,new_cords_y2)))
70
71 New_Binary_2=np.zeros((100,100))
72
73 New_Binary_2=np.asarray(New_Binary_2, dtype=np.bool)
74 New_Binary_2=New_Binary_2.astype(int)
75
76 New_Binary_2[Pinakas_cords2[:,0],Pinakas_cords2[:,1]]=1
77
78 plt.figure(3)
79 plt.imshow(New_Binary_2,cmap=plt.cm.gray)
80 plt.title('8-Gitiniasi_Image_Perimeter')
81

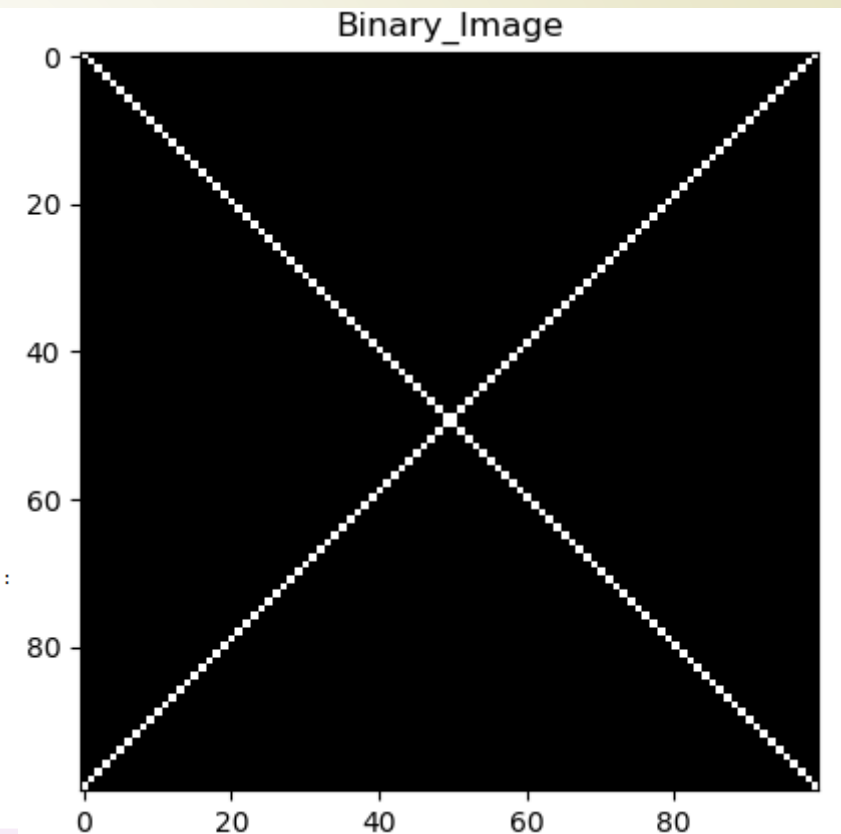
```



Δημιουργία εικόνας και αποθήκευση συντεταγμένων/οπτικοποίηση περιγράμματος 4-8 Γειτνίαση

11

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 Image_Binary=np.zeros((100,100))
5
6 sum1=0
7
8 for i in range(len(Image_Binary)):
9     for j in range(len(Image_Binary)):
10         if(i==j):
11             Image_Binary[i,j]=1
12
13 for j in reversed(range(len(Image_Binary))):
14
15     Image_Binary[sum1,j]=1
16     sum1=sum1+1
17
18 Image_Binary=np.asarray(Image_Binary, dtype=np.bool)
19 Image_Binary=Image_Binary.astype(int)
20
21 plt.figure(1)
22 plt.imshow(Image_Binary,cmap=plt.cm.gray)
23 plt.title('Binary_Image')
24
25 cords_x=[]
26 cords_y=[]
27 for i in range(len(Image_Binary)-1):
28     for j in range(len(Image_Binary)-1):
29         if (Image_Binary[i,j]==1 and (Image_Binary[i,j-1]==0 or Image_Binary[i,j+1]==0 or Image_Binary[i-1,j]==0 or Image_Binary[i+1,j]==0)):
30             cords_x.append(i)
31             cords_y.append(j)
32
33 new_cords_x=[]
34 new_cords_y=[]
35 New_Binary=Image_Binary
36 for i in range(len(New_Binary)-1):
37     for j in range(len(New_Binary)-1):
38         if (New_Binary[i,j]==1 and (New_Binary[i,j-1]==0 or New_Binary[i,j+1]==0 or New_Binary[i-1,j]==0 or New_Binary[i+1,j]==0
39         or New_Binary[i-1,j-1]==0 or New_Binary[i-1,j+1]==0 or New_Binary[i+1,j-1]==0 or New_Binary[i+1,j+1]==0)):
40             new_cords_x.append(i)
41             new_cords_y.append(j)
42 |
```



Challenge ?

► Open your project_1 script and:

1. Show ONLY the BINARY body of the mermaid
2. Find its perimeter and then depict it
3. Subplot these two images into one figure

