

# HW 6

b05902019 資工三 蔡青邑



Using "lena.bmp" as input image.

## Python Packages I used

- `skimage.io`: for basic image i/o.
- `numpy`: for the convenience of array manipulation.

## Some Other Functions I Build

- **`binarize(img, lower_expand, upper_expand, threshold)`**: binarize the image(`img`) according to the threshold and return it.
- **`yokoi(f)`**: given the four type in `f` (i.e. `['r', 'r', 'r', 'r']`), returning the corresponding yokoi number.

# How I Implemented

1. After reading the input image, I binarized it.
2. Then followed by the requirement, I shrink the image into 64 x 64.

```
height = int(height / 8)
width = int(width / 8)
for i in range(height + 2):
    frame.append([])
    for j in range(width + 2):
        frame[-1].append(0)
```

3. Then I output the yokoi number according to the formulas and definition of our slide with my self-defined dunction, `yoikoi()`.

$$h(b, c, d, e) = \begin{cases} q & \text{if } b = c \text{ and } (d \neq b \vee e \neq b) \\ r & \text{if } b = c \text{ and } (d = b \wedge e = b) \\ s & \text{if } b \neq c \end{cases}$$
$$f(a_1, a_2, a_3, a_4) = \begin{cases} 5 & \text{if } a_1 = a_2 = a_3 = a_4 = r \\ n & \text{where } n = \text{number of } \{a_k | a_k = q\}, \text{ otherwise} \end{cases}$$

```
ans = [[]]
delta_c = np.array([[1, 0], [0, 1], [-1, 0], [0, -1]])
delta_d = np.array([[1, 1], [-1, 1], [-1, -1], [1, -1]])
delta_e = np.array([[0, 1], [-1, 0], [0, -1], [1, 0]])
for i in range(1, height + 2 - 1):
    for j in range(1, width + 2 - 1):
        if(frame[i][j] == 0):
            print(' ', end = '')
        else:
            f = []
            for d in range(4):
                Type = 'chiu'
                o = np.array([i, j])
                b, c, d, e = o, o + delta_c[d], o + delta_d[d], o +
                delta_e[d]
                # print(b, c, d, e)
                b, c, d, e = frame[b[0]][b[1]], frame[c[0]][c[1]],
                frame[d[0]][d[1]], frame[e[0]][e[1]]
                if(b == c and (d != b or e != b)):
                    Type = 'q'
                elif(b == c):
                    Type = 'r'
                else:
                    Type = 's'
                f.append(Type)
            ans = yokoi(f)
            print(' ' if(ans == 0) else ans, end = '')
print()
```

## The Result I Get

11111111	121111111111122322221	1111111111111
15555551	115555555511 2 11 11	1155555555511
15555551	1 2115555112 21112221	155555555551 21
15555551	1 2 155112 22221511	1555555555511 1
15555551	22 2112 22 121	15555555555511
15555551	1 2 21 2 1 1	15555555555551
15555551	12 1 121111 1321	155555555555511
15111551	1322 1155551111	155555555555551
111 1551	1 121555555511	155555555555511
11 1551	211555555511	15511155555511
21 1551	2 15555555111	1551 11555511
1 1551	2 155555555511	1551 115551 1
1551	112115555555551	1551 15511 12
1551	1555555555555511	1551 1111 111
1551	1 222115555555555511	1151 11 1151
1551	2 22 1 155555555555511	151 11111 1551
1551	2 1 1155555555555551	151 115551 11551
1551	2 1155555555555555111511155511	115551
1551	12 115555555555555555555551	155551
1551	11 22155555555555555555555112	1155551
1551	111 22 1555555555555555555551 1	1555551
1551	1511 1 12511211111211155555555111	11555551
1551	15521 1 121 1 11 1 1555555111	15555551
1551	1151 132 2 1155555111	115555551
1551	151 322 115555111 121	155555551
1551	1221 2 1555551 131	115555551
1551	2 1 115555511 1	1155555551
1551	2 115555551	1 155555551
1551	2 1155555551	21155555551
1551	1 11555555551	15555555551
1551	1 11511115555521 1	115555555551
1551	1 1 11111 1155511 2	155555555551
1551	131 111 15111 2	155555555551
1551	121 1121 1 111 1 2	1155555555551
1551	11 111 1 221 11 1 2	1555555555551
1551	12 1 21 121 11 1111 2	15555555555551
1551	1 12 22 151111111551 2	11555555555551
1551	1 2 1555551115511 1	15555555555551
1551	2 22 12555551 15551 1	15555555555551
1551	1 1 1555511 11511 2	115555555555551
1551	21 155551 1 151 2	155555555555551
1551	2 15555112 151 2	155555555555551
1551	1 1 1 115555511111 2	155555555555551
1551	2 22 111511111212 2115555555555551	
1551	1 12 151 2 1 15555555111555551	
1551	1111 121 155555551 1555551	
1551	11111111 155555551 1555551	
1551	115551 155555551 1555511	
1551	15551 211111111 155511	
11521	1 12 122155511 2 11 115511	
1 151	1 1 155555111 2111 15511	
22 1511	1 15555555111 155111 1511	

22	1511	1	15555555551	155551	1151
2	151	1	11155555555511	155511	1511
2	1521	1	155555555555511	15551	12151
2	151	121	155555555555551	155511	1551
2	1511		1555555555555551	115551	1511
21	1511	11	1555555555555551	111111151	
11	151		11555555555555511	111511	
11	151		1555555555555551	151	
11	151		11555555555555551	211	
11	151		115555555555555511	1	
11	151		15555555555555551		
11	111		1211111111111111111		