

執行環境: python3, PIL, numpy

檔案:

hw6.py	程式碼
binarized.bmp	輸入圖
output.txt	輸出檔案
output.pdf	A4 格式的輸出檔

程式說明:

根據公式

- for 4-connectivity
  - $$h(b, c, d, e) = \begin{cases} q & \text{if } b = c \text{ and } (d \neq b \vee e \neq b) \\ r & \text{if } b = c \text{ and } (d = b \wedge e = b) \\ s & \text{if } b \neq c \end{cases}$$
  - $$f(b, c, d, e) = \begin{cases} 5 & \text{if } a_1 = a_2 = a_3 = a_4 = r \\ n & \text{where } n = \# \{a_k | a_k = q\}, \text{ otherwise} \end{cases}$$
- $a_1 = h(x_0, x_1, x_6, x_2)$   
 $a_2 = h(x_0, x_2, x_7, x_3)$   
 $a_3 = h(x_0, x_3, x_8, x_4)$   
 $a_4 = h(x_0, x_4, x_5, x_1)$
- $output = f(a_1, a_2, a_3, a_4)$

實際實作時將界外的點值設為-1

```
def val(pos):
    r, c = pos
    return -1 if r < 0 or c < 0 or r >= R or c >= C else arr[r][c]

def h(a, b, c, d):
    va, vb, vc, vd = val(a), val(b), val(c), val(d)
    if va != vb: return 's'
    return 'r' if vb == vc and vc == vd else 'q'

def f(r, c):
    ret = [h((r, c), (r, c+1), (r-1, c+1), (r-1, c)), \
           h((r, c), (r-1, c), (r-1, c-1), (r, c-1)), \
           h((r, c), (r, c-1), (r+1, c-1), (r+1, c)), \
           h((r, c), (r+1, c), (r+1, c+1), (r, c+1))]
    return 5 if (ret.count('r')==4) else ret.count('q')
```

最後輸出時只印出亮度為 255 的 pixel 的 yokoi number ,

亮度為 0 的 pixel 則以空格代替

```
file = open('output.txt', 'w')
for r in range(R):
    if r != 0: file.write('\n')
    for c in range(C):
        tmp = ' ' if arr[r][c] == 0 else f(r, c)
        file.write(' ' if tmp == 0 else str(tmp))
file.close()
```

結果:

output.txt

111111111	1211111111111122322221	111111111111111
15555551	115555555511 2 11 11	1155555555511
15555551	1 2115555112 21112221	15555555551 21
15555551	1 2 155112 22221511	1555555555511 1
15555551	22 2112 22 121	1555555555511
15555551	1 2 21 2 1	1555555555551
15555551	12 1 121111 1321	15555555555511
15111551	1322 1155551111	15555555555551
111 1551	1 12155555511	15555555555511
11 1551	2115555511	155115555511
21 1551	2 1555555111	1551 1155511
1 1551	2 15555555511	1551 115551 1
1551	11211555555551	1551 15511 12
1551	15555555555511	1551 1111 111
1551	22211555555555511	1151 11 1151
1551	22 1 155555555555511	151 11111 1551
1551	2 1 115555555555551	151 11551 11551
1551	2 115555555555551151115511	1155115551 11551
1551	12 115555555555555555555551	155551 155551
1551	11 22155555555555555555555112	1155551 1155551
1551	111 22 15555555555555555551 1	1555551 1555551
1551	1511 1 1251121111121115555555111	11555551 11555551
1551	15521 1 121 1 11 1 1555555111	15555551 15555551
1551	1151 132 2 1155555111	115555551 115555551
1551	151 322 11555111 121	155555551 155555551
1551	1221 2 155551 131	1155555551 1155555551
1551	2 1 11555511 1	1155555551 1155555551
1551	2 115555551	1 155555551
1551	2 1155555551	2115555551
1551	1 11555555551	15555555551
1551	1 11511115555521 1	115555555551
1551	1 1111 1155511 2	155555555551
1551	131 111 15111 2	1555555555551
1551	121 1121 1 111 1 2	11555555555551
1551	11 111 1 221 11 1 2	15555555555551
1551	12 121 11 1111 2	15555555555551
1551	1 12 22 151111111551 2	115555555555551
1551	1 2 155555115511 1	155555555555551
1551	2 22 1255551 1551 1	155555555555551
1551	1 1555511 11511 2	1155555555555551
1551	21 155551 1 151 2	1555555555555551
1551	2 1555512 151 2	1555555555555551
1551	1 1 115555511111 2	1555555555555551
1551	2 22 111511111212 2115555555555551	1555555555555551
1551	1 12 151 2 1 1555555111555551	1555555551 1555551
1551	1111 121 15555551 1555551	1555555551 1555551
1551	11111111 15555551 15555511	2111111111 155511
1551	15551 2111111111 115511	2 11 115511
11521	1 12 122155511 2111 15511	1 1511
1 151	1 1 155555111 155111 1511	155551 1151
22 1511	1 1555555111 155551 1511	155555555511 155511 1511
22 1511	1 1555555551 155551 1511	155551 12151
2 151	1 115555555551 155511 1511	15555555555551 155511 1551
2 1521	1 15555555555551 15551 1511	155551 1511
2 151	121 15555555555551 1111111151	1111111151
2 1511	11 155555555555551 111511	151
21 1511	11 1555555555555551 211	1
11 151	11555555555555551 1	
11 151	115555555555555551	
11 151	115555555555555551	
11 151	155555555555555551	
11 111	12111111111111111111	