Dilation

Erosion

Opening

Closing

(a) Dilation

```
def dilation(img,kernel):
    im = np.array(img)
    temp = np.zeros((512,512))
    for i in range(width):
        for j in range(height):
            if im[i][j] > 0:
                max_v = 0
                for point in kernel:
                    x, y = point
                    if (x+i) >= 0 and (x+i) < width and (y+j) >= 0 and (y+j) < height:
                        if im[x+i][y+j] > max_v:
                            max_v = im[i+x][j+y]
                for point in kernel:
                    x,y = point
                    if (x+i) >= 0 and (x+i) < width and (y+j) >= 0 and (y+j) < height:
                        temp[x+i][y+j] = max_v
    result = Image.fromarray(np.uint8(temp))
    result.show()
    return result
```

先讀入原始灰階照片後轉為 ndarray

利用迴圈為找出 kernel 中的 local maximum

再利用第二個迴圈將各 pixel 值轉為 maximum value

(b) Erosion

```
def erosion(img, kernel):
    im = np.array(img)
    temp = np.zeros((512,512))
    for i in range(width):
        for j in range(height):
            if im[i][j] > 0:
                exist = True
                min_v = 512
                for point in kernel:
                    x, y = point
                    if (x+i) >= 0 and (x+i) < width and (y+j) >= 0 and (y+j) < height:
                        if im[x+i][y+j] == 0:
                            exist = False
                            break
                        if im[x+i][y+j] < min_v:
                            min_v = im[x+i][y+j]
                    else:
                        exist = False
                        break
                if exist:
                    temp[x+i][y+j] = min_v
    result = Image.fromarray(np.uint8(temp))
    # result.show()
    return result
```

先讀入原始灰階照片後轉為 ndarray
接著利用迴圈判斷 kernel 是否能 fit 原始的 pixel（除了黑色以外）
若能完全 fit 則判斷 kernel 中的 local minimum
最後再將各個 pixel 值轉為 local minimum

(c) Opening & Closing

```
def opening(img, kernel):
    return dilation(erosion(img, kernel), kernel)

def closing(img, kernel):
    return erosion(dilation(img, kernel), kernel)
```

According to the course slides,

The definition of opening is $B \circ K = (B \ominus K) \oplus K$

The definition of closing is $B \bullet K = (B \oplus K) \ominus K$