

1. 遍歷每個像素點，將像素點項轉成 grayscale image，再轉成 binary

image (與作業一的方式相同)。

2. 判斷此像素點是否為 foreground，不是則略過，若是的話，到步驟 3。

3. 分為 4-connected 以及 8-connected。

label list 紀錄每個像素點被 label 的值 (預設給 0)。

label_p 紀錄目前 object 的數量 (預設給 0，會從 1 開始計)

紀錄兩個值為同一個 object 的方法：使用 union find 的資料結構紀錄。

甲、4-connected

- i. a 為此像素點左方的 label，b 為此像素點上方的 label。若此像素點的左方或上沒有像素點則給 0。
- ii. 若 a, b 同時皆沒有被 label 過 (皆為 0)，則 $\text{label_p} += 1$ ，把 label_p 給此像素點位置的 label。
- iii. 若 a, b 只有其中一個被 label 過，則把被 label 過的值給此像素點位置的 label。
- iv. 若 a, b 兩個都被 label 過，若 a, b 值相同，則把 a 或 b 任一個值給此像素點位置的 label。若 a, b 值不同，則把比較小的值給此像素點位置的 label，並紀錄這兩個值為同一個 object。

乙、8-connected

發現有更簡潔的做法，也可以用到 4-connected。

i. a 為此像素點左方的 label，b 為此像素點左方的上方的 label，c 為此像素點上方的 label，d 為此像素點上方右方的 label。若沒有像素點則給 0。

ii. 若 a, b, c, d 同時皆沒有被 label 過 (皆為 0)，則 $\text{label_p} += 1$ ，把 label_p 給此像素點位置的 label。

iii. 若 a, b, c, d 有被 label 過，把有被 label 過中的最小值給此像素點位置的 label。再把最小值與其他的值紀錄為同一個 object，若值相同則略過。

4. 產生 $\text{label_p}+1$ 個不重複的顏色 (因 label_p 也是一個 index)。

5. 最後再填入顏色，遍歷每個像素點，若為前景，則尋找此 label 值的 parent 值，再依據這個 parent 的值，從 color 找 index 是 parent 的顏色給新圖像中的像素點。若為 background，則把 255 (白色) 給新圖像中的像素點。



img1_4

img1_8



img2_4

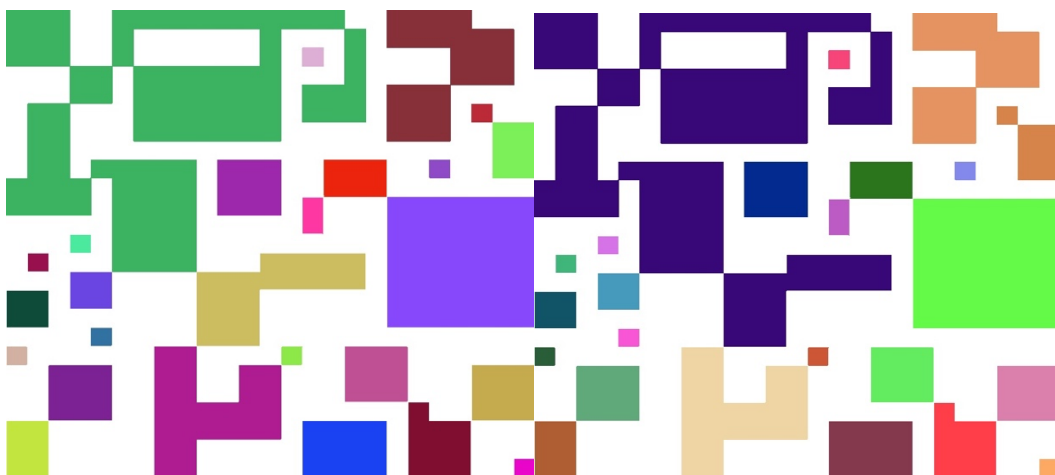


img2_8



Img3_4

img3_8



Img4_4

img4_8