Anthony Christe
ICS 683
Tuesday, September 24th

Homework 2

1) Original Image

	1	1			1			1	
1	1				1			1	
1		1	1	1	1	1	1		1
1						1	1		
1		1	1	1					
1		1		1		1		1	
1		1		1		1	1	1	
1		1	1	1		1	1	1	
1							1		
1	1	1	1	1	1	1	1		

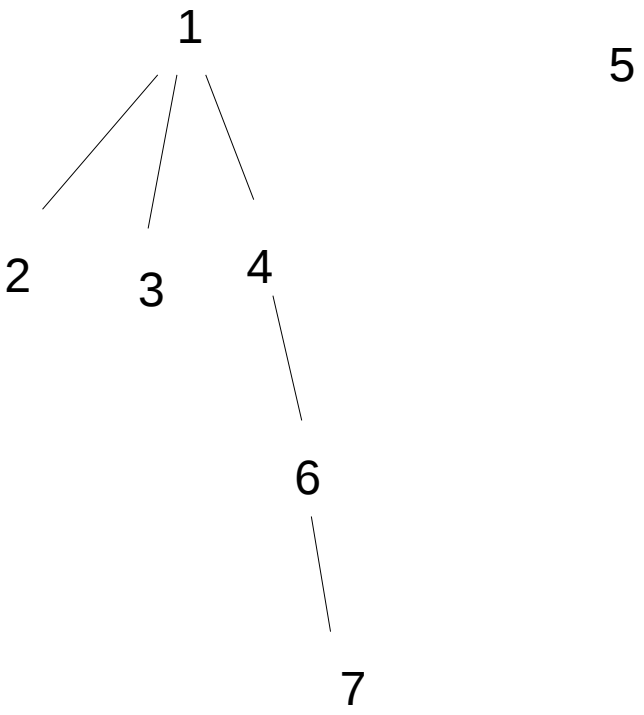
A) Temporary labels after first pass.

	1	1			2			3	
4	1				2			3	
4		1	1	1	1	1	1		3
4						1	1		
4		5	5	5					
4		5		5		6		7	
4		5		5		6	6	6	
4		5	5	5		6	6	6	
4							6		
4	4	4	4	4	4	4	4		

B) Union-Find Data Structure

1	2	3	4	5	6	7
0	1	1	1	0	4	6

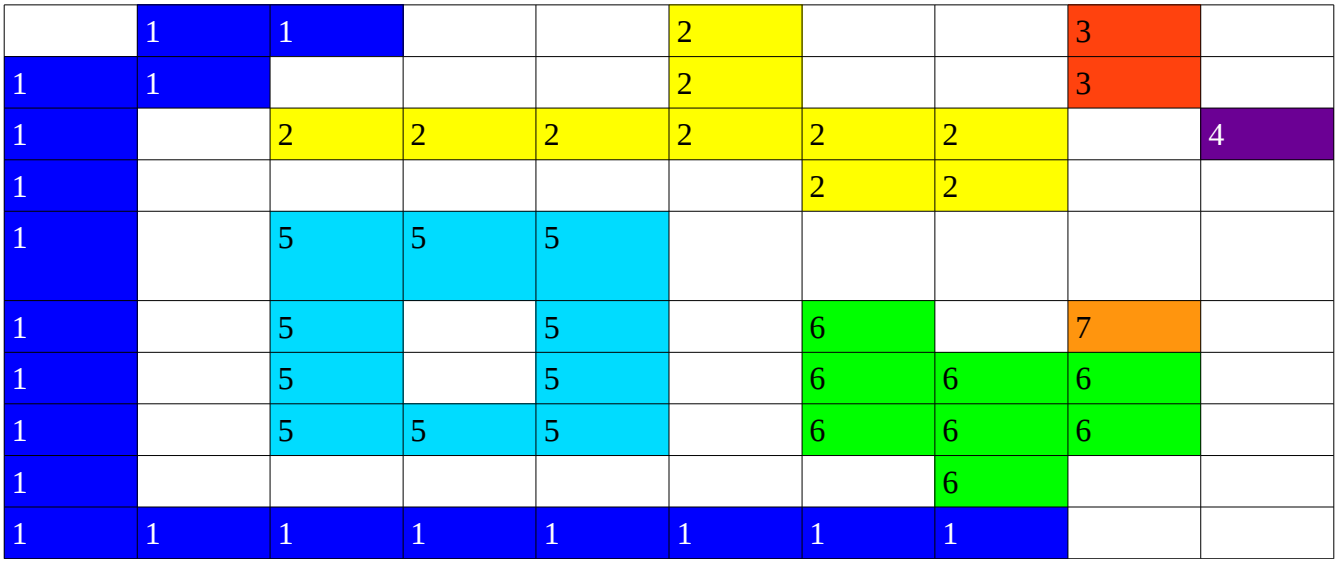
Or looked at in another way:



C. Final Image

	1	1			1			1	
1	1				1			1	
1		1	1	1	1	1	1		1
1						1	1		
1		5	5	5					
1		5		5		1		1	
1		5		5		1	1	1	
1		5	5	5		1	1	1	
1							1		
1	1	1	1	1	1	1	1		

2) Run-length Encoding
A) First Pass



Run-length Data Structure – Row Index

row	rtart	rend
0	1	3
1	4	6
2	7	9
3	10	11
4	12	13
5	14	18
6	20	22
7	23	25
8	26	27
9	28	28

Run-length Data Structure - Runs

run	row	scol	ecol
1	0	1	2
2	0	5	5
3	0	8	8
4	1	0	1
5	1	5	5
6	1	8	8
7	2	0	0
8	2	2	7
9	2	9	9
10	3	0	0
11	3	6	7
12	4	0	0
13	4	2	4
14	5	0	0
15	5	2	2
16	5	4	4
17	5	6	6
18	5	8	8
19	6	0	0
20	6	2	2
21	6	4	4
22	6	6	8
23	7	0	0
24	7	2	4
25	7	6	8
26	8	0	0
27	8	7	7
28	9	0	7

B) Union-find Data Structure

1	2	3	4	5	6	7
0	0	0	0	0	1	6

I'm concerned that I may have misunderstood the connected component algorithm, but it appears that the union method is only called when run-p and run-q are overlapping with different parent labels. This only happens twice with 6,1 and 7,6.

I couldn't find any examples in the lecture notes which clearly made the correct distinction between overlapping and non-overlapping at the border case. Perhaps I'm wrong, but I interpreted not overlapping to mean 4-neighbor connectivity, so if two runs are exactly diagonal to each other, that is not considered overlapping.

For example, I consider the following runs to be non-overlapping:
Q
***
    ****
      P

C) Final Image

	1	1			2			3	
1	1				2			3	
1		2	2	2	2	2	2		4
1						2	2		
1		5	5	5					
1		5		5		1		1	
1		5		5		1	1	1	
1		5	5	5		1	1	1	
1							1		
1	1	1	1	1	1	1	1		

2) Mathematical Morphology and Coins
First apply opening to the image to smooth the ridges along the edge of the coins.

Q) Finding the Quarter
I = The original image.

DQ = Dilate the image with with disk structuring element with radius of a penny plus 1 so that the disk SE is just slightly larger than the penny.

This causes the quarter and only the quarter to grow in size. The penny and dime are unchanged.

Q1 = DQ - I

By subtracting the original image from the dilated image, only one component with a hole inside of it will remain. The hole represents the quarter, but it currently represented by background pixels surrounded by foreground pixels from the dilation.

Since we've completely eliminated the penny and the dime, it's known that the quarter is the only component left and we can obtain the bounding box of the dilated quarter.

Q = (not Q1) and boundingbox(I)

The quarter is found by flipping all the bits in the subtracted image so that the hole becomes the foreground. We then AND this with the original image only within the bounding box, so that the surrounding pixels created from the dilation are set to background pixels.

P) Finding the Penny
PI = I - Q

To start with the penny, we subtract the quarter image from the original image to create an image that only contains the penny and the dime.

We then repeat the same process we used to find the quarter, except this time we use a structuring element with a diameter just slightly larger than the dimes. When the process finishes, only the penny, P, is left.

D) Finding the Dime
D = PI – P

To find the dime, simply subtract from the image with a penny and a dime, the image of just the penny. This will ensure that only the dime remains.

Notes on this method
This was the best solution I could come up with, but I realize the use of the bounding box might be discouraged. I wasn't able to come up with a good solution without using the bounding box.

If I were forced to not use a bounding box, I would attempt to do something like the following:

- Dilate image with SE slightly larger than penny.
- Subtract the original image from the dilated image.
- A component with a hole will remain representing the quarter.
- Apply closing until the hole is filled in.
- Erode component until it is same size as quarter.
- Rinse and repeat with other coins subtracting the previous result for each image.

However, I feel that the hole left by the subtraction of the original image and the dilated image will be too large to successfully close with closing. Eroding the image until it's the same size feels more like hand waiving than an actual solution. For those reasons, I chose to solve this with the assumption of being able to use a bounding box.