Keone Hiraide
September 23, 2013
ICS 683

Homework #2

1. Consider the following binary 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		

(1) (15 points) Apply the classical connected component labeling algorithm with the union-find data structure assuming 8-neighbor connectivity.

A. Show the temporary labels after the first pass. When there is a conflict, i.e. two different labels can propagate to the same pixel, assign the smaller label.

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

B. What are the equivalences between temporary labels recorded during the first pass?
Set 1: 1,2,3,5,6
Set 2: 4

C. Show the final labels after replacing temporary labels with equivalence labels.

	1	1			1			1	
1	1				1			1	
1		1	1	1	1	1	1		1
1						1	1		
1		4	4	4					
1		4		4		1		1	
1		4		4		1	1	1	
1		4	4	4		1	1	1	
1							1		
1	1	1	1	1	1	1	1		

(**2) (15 points)**

Apply the classical connected component labeling algorithm with the run-length encoding.

A. Show the temporary labels after the first pass.

4-Connectivity was used to form the first pass with run-encoding.

	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		6		7	
1		5		5		7	7	7	
1		5	5	5		7	7	7	
1							7		
7	7	7	7	7	7	7	7		

B. What are the equivalences between temporary labels recorded during the first pass?

Set 1: 1, 6, 7
Set 2: 2
Set 3: 3
Set 4: 4
Set 5: 5

C. Show the final labels after replacing temporary labels with equivalence labels.

	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. (20 points)
A camera takes an image I of a penny, a dime, and a quarter lying on a white background and not touching one another. Thresholding is used successfully to create a binary image B with 1 bits for the coin regions and 0 bits for the background. You are given the known diameters of the coins D_P, D_D, D_Q. (Note that $D_Q > D_P > D_D$.) Using the operations of mathematical morphology (dilation, erosion, opening, closing) and the logical operators (e.g. AND, OR, NOT, and MINUS (set difference)), show how to produce three binary output images: P, D, and Q. P should contain just the penny (as 1 bits), D should contain just the dime, and Q should contain just the quarter. (Assume that the foreground regions (regions of 1 bits) in the binary image cover each coin region completely.)

- Use opening on image I where the structural element would have diameter D_Q. The output would be binary image Q.

- I MINUS Q will result an image I_2. Use opening on image I_2 where the structural element would have a diameter D_P. The output would be binary Image P.

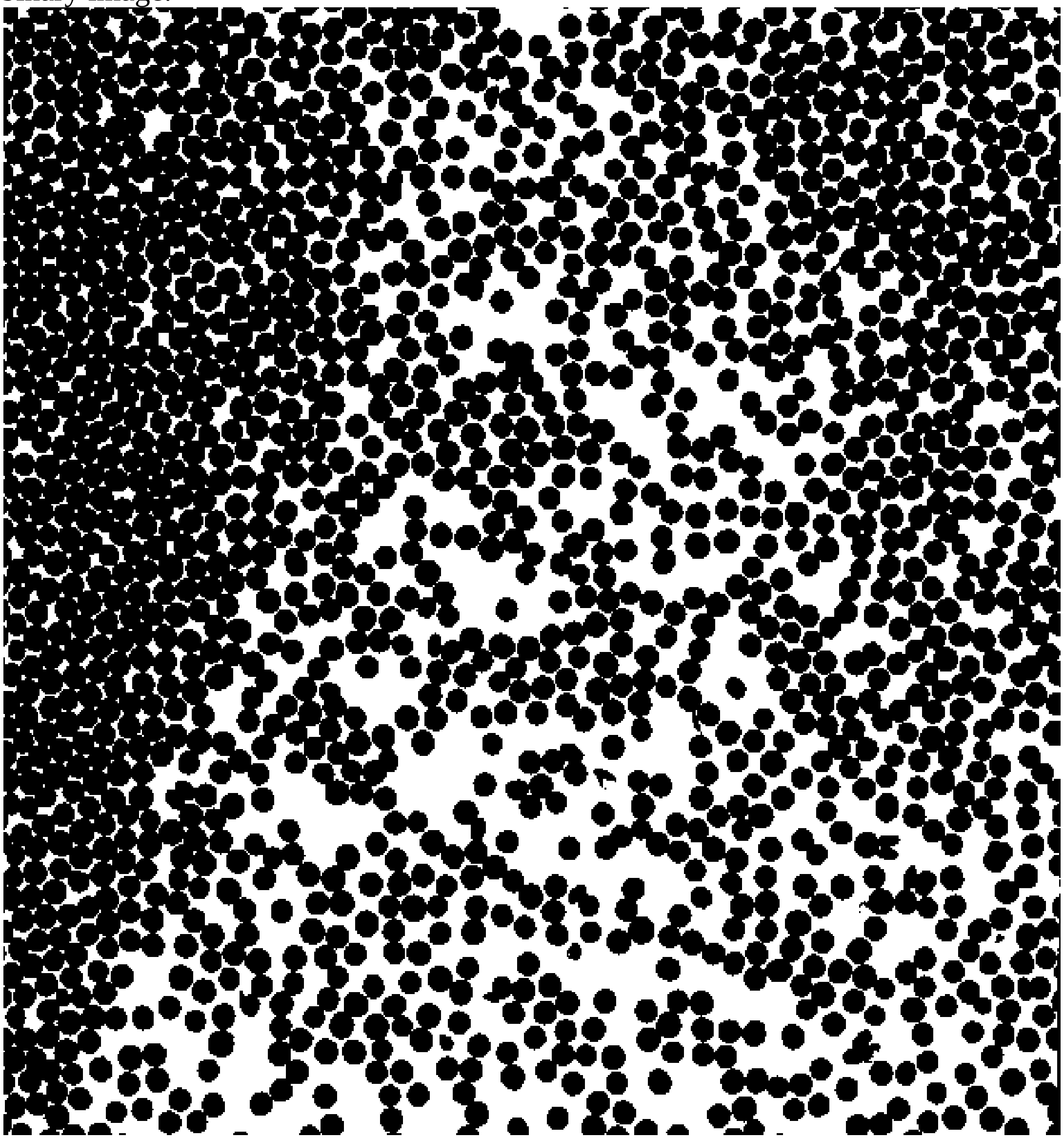
- I_2 MINUS P would result in the output being binary image D.

3. Download the image called “balls.gif” from the course website. Write a program to determine how many distinct balls are in the image as follows.

(1) (10 points) Implement Otsu’s Algorithm to find a global threshold. What is the threshold value found?

Creating a histogram from “balls.gif” resulted in 99 bin slots for my histogram. Inputting the histogram into my Otsu Algorithm implementation resulted in t = 54 with the threshold value being 103. Note that the greyscale value of each pixel used is from 0-255.

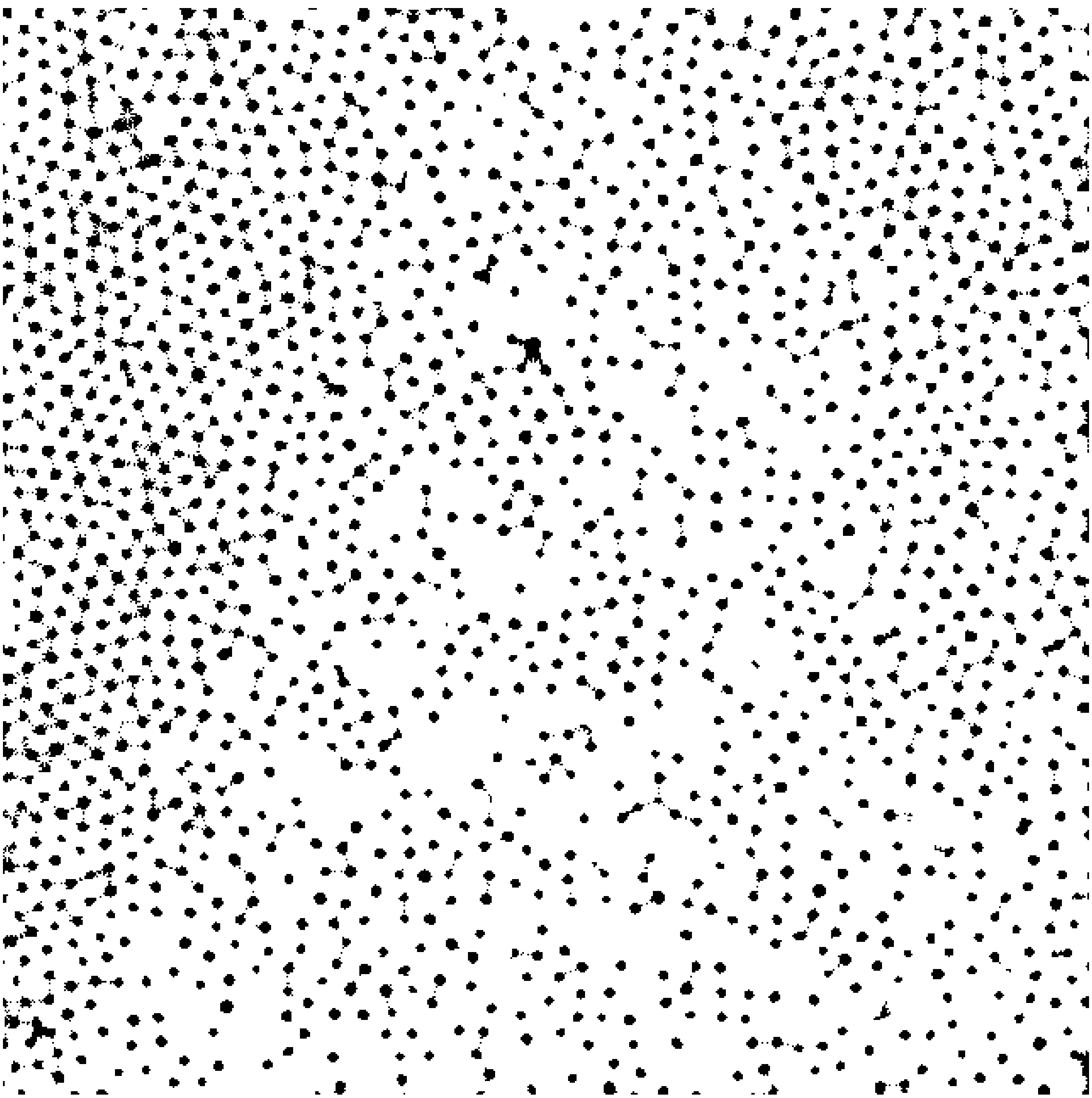
(2)(10 points)
Using the threshold value found in (1), convert the input grayscale image to a binary image in which the balls are foreground regions. (Note that the balls are dark blobs in the original image.) Show the resulting binary image.



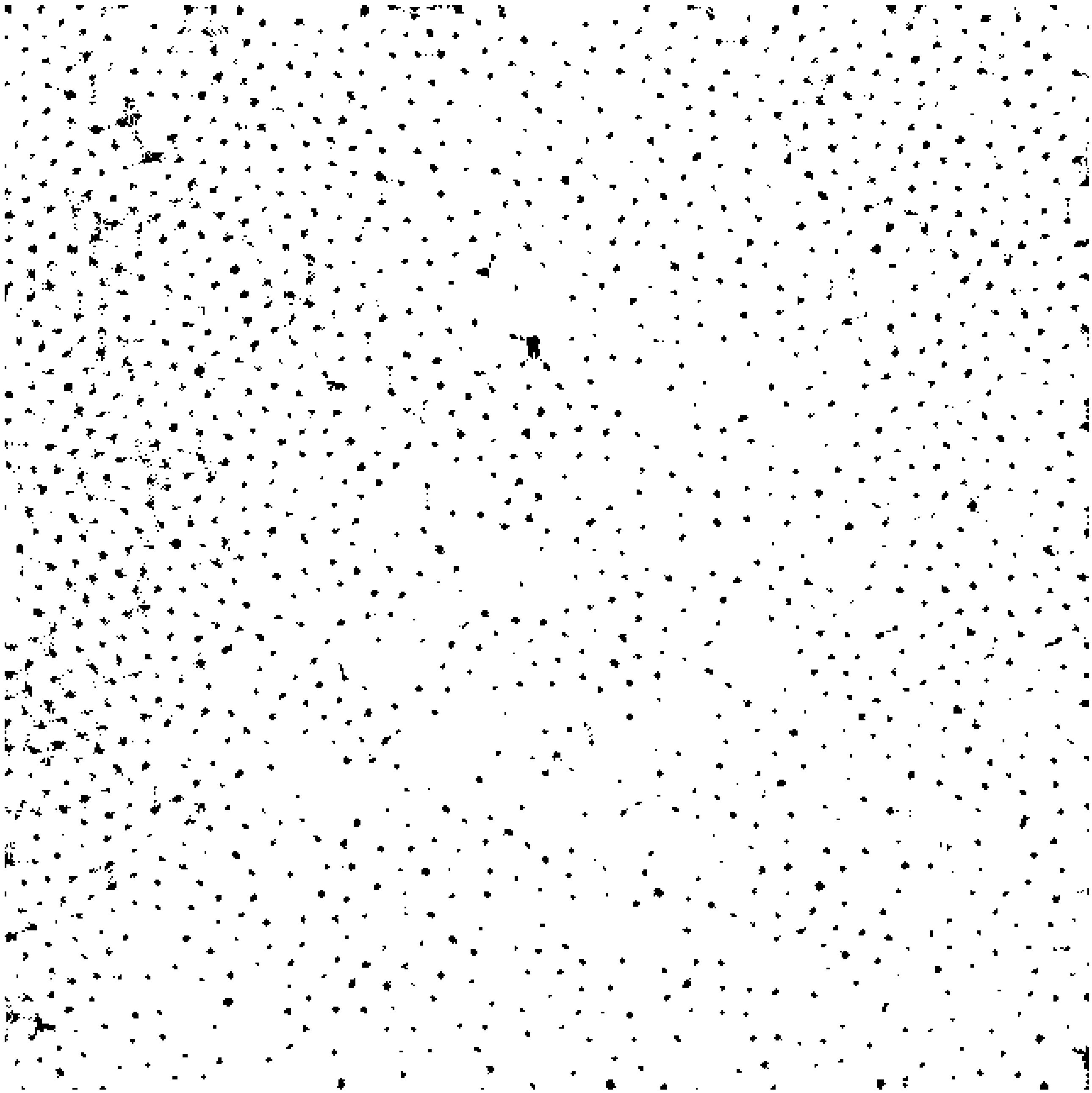
(3) (15 points) Create disc structuring elements (SEs) with radius 4, 5, and 6, respectively. Erode the ball regions in the binary image from (1) using those disc SEs and show the resulting images. How are they different? What size of the disc SE do you think gives the best result and why?

In regards to “best results”, it really depends on what you're trying to accomplish. An SE with a Radius of 4 erodes just enough so that you can still see the areas where the balls were connected in the original image. If your goal was to spot which balls overlapped, radius 4 would work well. If you wanted to identify the number of balls from the original image, radius 5 would would be the best because most of the overlaps have been eroded, while still keeping the balls intact. I would say that radius 6 takes away too much information, and therefore, probably not a SE which fits well. In general, if you have a desk SE, the larger the radius, the number of pixels that will be eroded will increase.

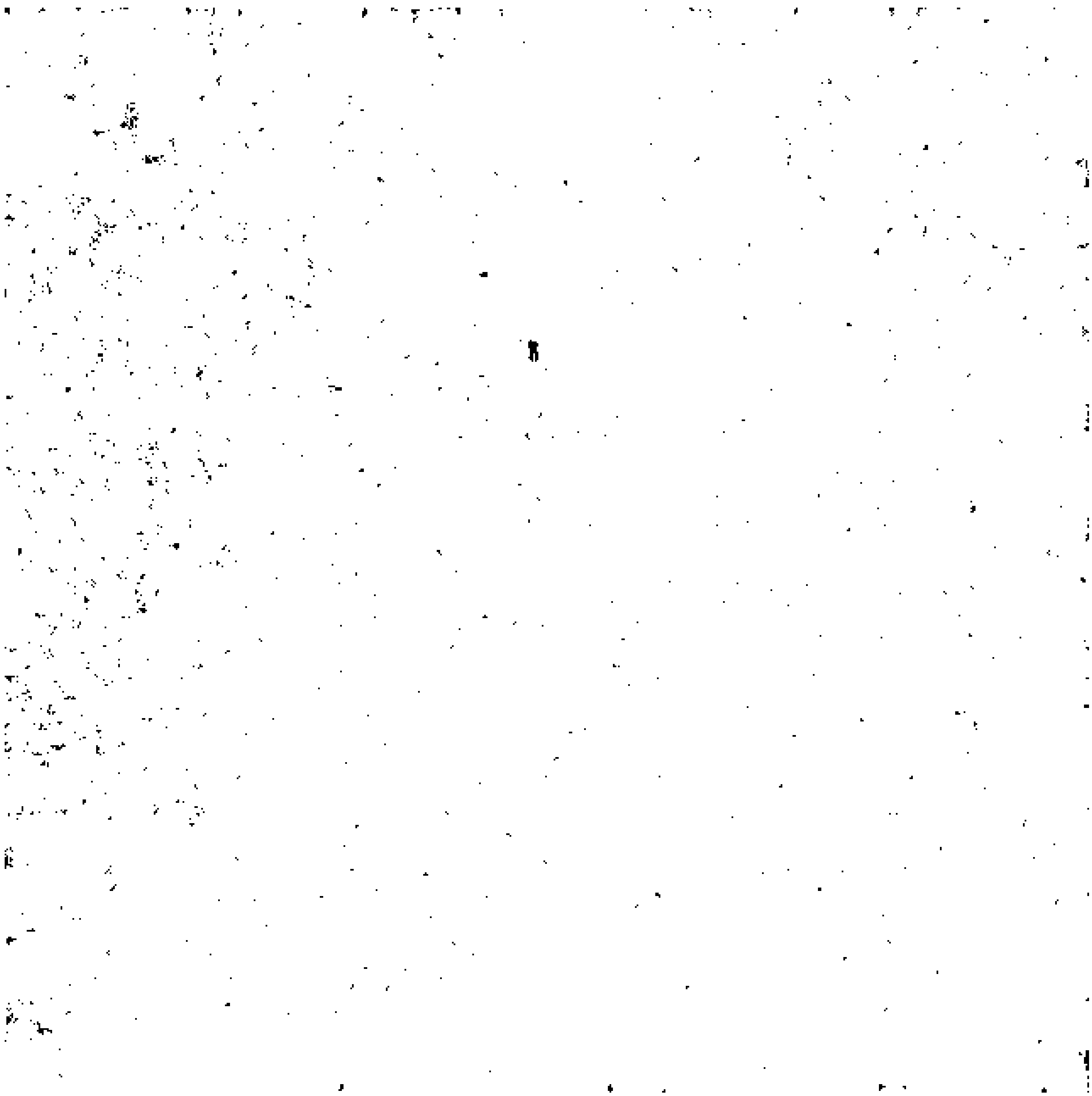
Radius 4:



Radius 5:



Radius 6:



(4) (15points) Implement connected component labeling with 4-neighbor connectivity and run it on the images from (3). What are the resulting counts of the number of connected components for the three cases?

Radius 4: 2567 connected components

Radius 5: 2303 connected components

Radius 6: 1047 connected components