Problem 1

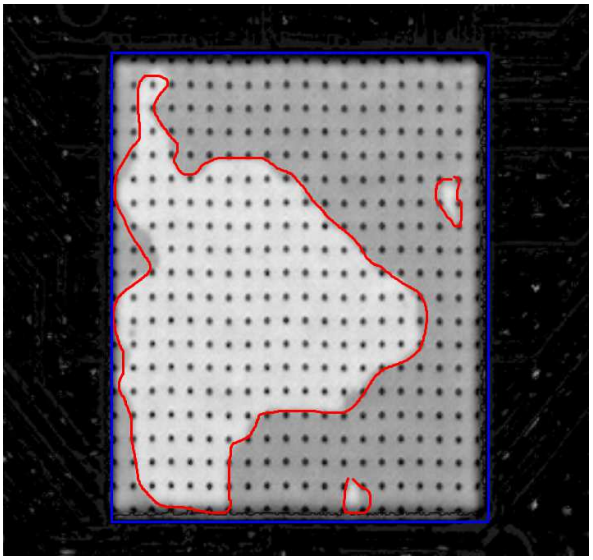
You are given an image of a defect (attached file Defect.png). Find the portion of gray rectangle that being covered by bright area.

In the image below, bright area is highlighted by the red line, rectangular area is highlighted by blue.

The portion = (area enclosed by red line) / (area enclosed by blue line)

Bonus: If you use some constants like for pixel intensity thresholding or for other purposes, please think about an automatic way to calculate those constants, to make algorithm generalizable for other defective images.

Note: Both colored lines are just references the real image doesn't have them.



Problem 2

Consider election process, you have two candidates X and Y.

You are in charge of vote forecast, and you created two graphs that correspond to social interaction between voters. The edge list for undirected Graph 1 and Graph 2 can be found in attachment. Node id is from 0 to 9999.

40% of nodes know they will vote for X no matter what

40% of nodes know they will vote for Y no matter what

20% of nodes are swing voters

Nodes with last digit in node id 0-3 (like 0,1,2,3,10,11,12,13, etc) support X no matter what

Nodes with last digit in node id 4-7 (like 4,5,6,7,14,15,16,17, etc) support Y no matter what

Nodes with last digit in node id 8-9 (like 8,9,18,19,28,29, etc) are swing voters

Only swing voters will go through 7-day “thought process”, based on majority of their friends.

1. Initialize graph with voters states (40 X, 40 Y, 20 swing).

The “thought process” is following:

2. For each iteration, swing voters are assigned a vote. If the greater number of friends of a swing voter support X, this swing voter will now support X. If the greater number of friends of a swing voter support Y, this swing voter will now support Y.
3. If friends’ votes are equally distributed between X and Y, the swing voter is assigned in alternating fashion, starting with X. This alternation is done at a global level for the whole network, across all rounds, i.e. the first tie is broken with X, the second with Y, and so on (not on a per node basis). You need to store a single global variable that keeps track of alternating vote, and initialize it to X in the first round. Then, as you iterate over nodes in order of increasing ID, whenever you assign a vote using this alternating variable, change its value afterwards.
4. When updating the swing votes, use the values from the current iteration. So, for example, when updating the votes for node 10, you should use the votes for nodes 0-9 from the current iteration, and nodes 11 and onwards from the previous iteration.

The “thought process” happens 7 times. On the eighth day, the votes are counted.

A. Basic Forecasting

Read both graphs. Assign initial vote configurations to the network. Then, simulate the voting process. Who will win in Graph 1, and by how many votes? Who will win in Graph 2, and by how many votes?

B. Winning with influential nodes

You want to win the election with Y, if we persuade people with the highest degree in the social graph to vote for Y, will it help?

You have 9000$.

Before the thought process, you will spend k dollars (1000, …,9000) on a cocktail party for influential people (have highest degree).

If you pay **k=1000**$, you invite one person (with the **most node degree**) to the party, and he/she will vote for Y no matter what. In case **of k=2000**$ you can invite the **two most influential person**, and they both will vote for Y. Etc…till k=9000. If several influential people have the same node degree, you break the ties such as smaller node ID goes first.

After spending k amount of money, we will go through the same thought process of counting votes.

Let’s examine how much you have to spend on cocktail party for candidate Y to win.

For each of the two graphs, please plot k (the amount you spend) on the x-axis (for values 1000, 2000, ..., 9000) and the number of votes you win by on the y-axis (this is a negative number if you lose).