

The 2020 Internship Software Challenge

Foreword

This document is designed to be, first and foremost, a test of your imagination.

As such, you are free to solve the challenge in any way you see fit, and using the programming language with which you are the most comfortable.

The input data for the various stages of the challenge has been included in the package, and the output data schema is loosely defined for each of the four stages.

Your task is to design software that connects given points in a matrix. You may recognize this as an automatic routing program.

There are four stages to this challenge, and you may solve however many you want from these four.

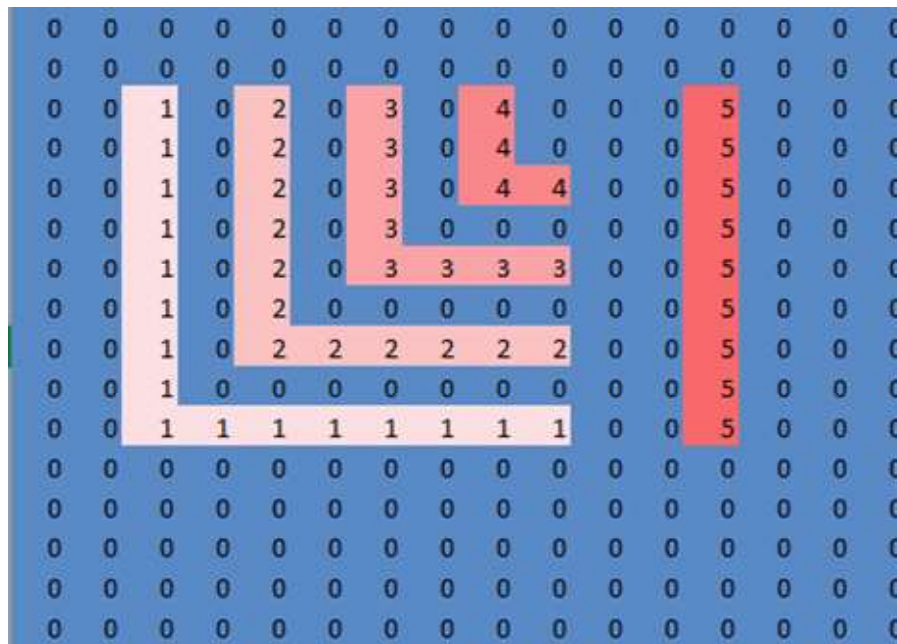
Step One

The input file for this step is called "Step_One.csv". It's a small matrix which, when represented in a spreadsheet editor, looks like this:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	2	0	3	0	4	0	0	0	5	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	5	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The goal is simple: write software that connects each pin with all the other pins that share the same value.

For example, and keep in mind that this is just one possible solution, the connections for this step may look like this:

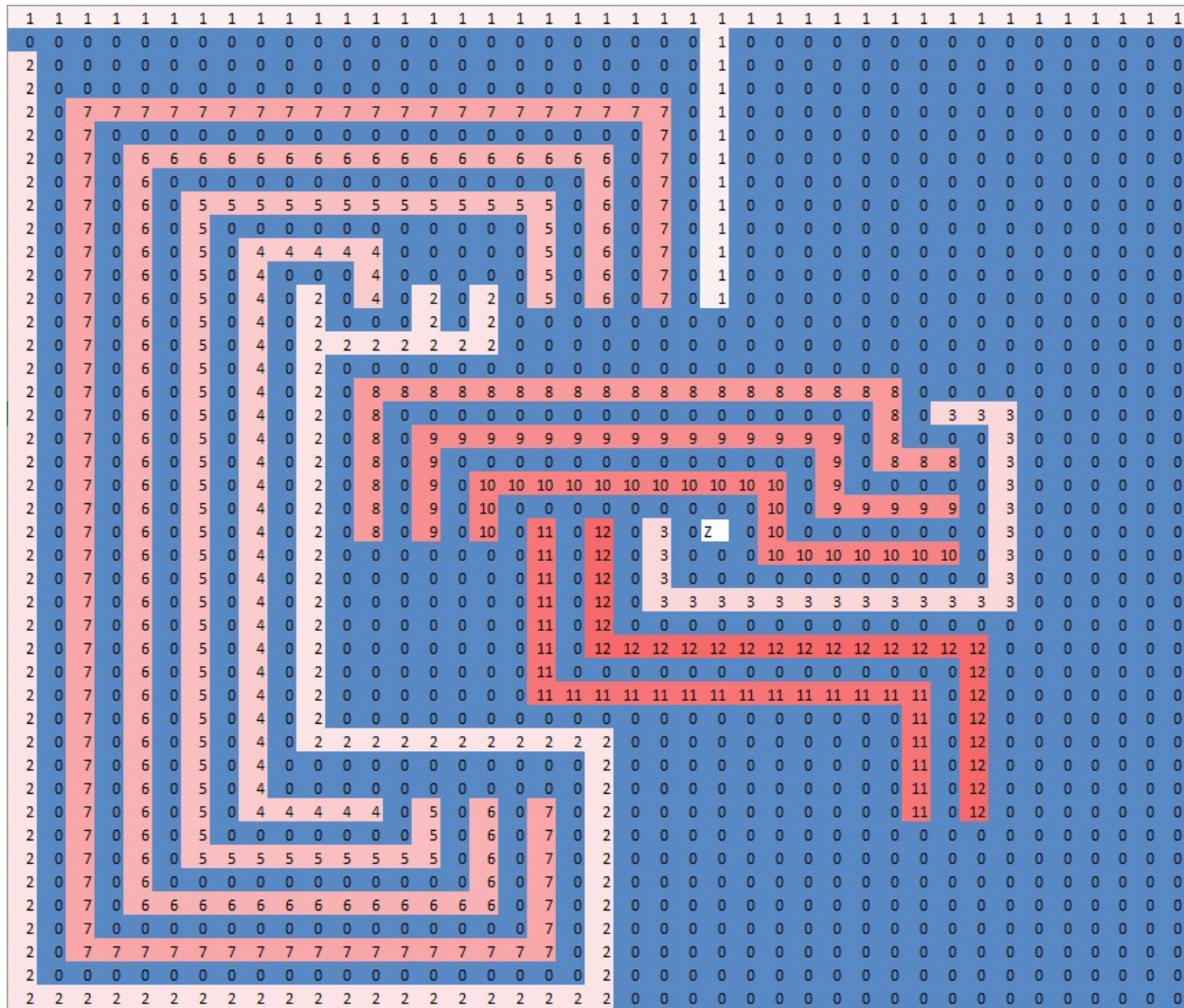


Some remarks:

- First and foremost, the connection traces must not touch or intersect. By this, we mean that there must always be at least one blank space between any given point in a connection trace and its nearest neighboring trace (orthogonally and diagonally). All the pins in the challenge input files are spaced so that there is at least one cell with a value of zero between any pin and its nearest neighbor, so your traces should follow the same rule.
- The traced routes must be contiguous.
- Also, the space in which to route these traces is limited. The size of the matrix in the input files is the maximum size for your playground. This is not an issue for step one, as the routing is simple. However, step two is a different story.
- You are not obligated to route the traces in a rectangular fashion. Diagonal or circular routing is a valid possibility, if you are brave enough. Diagonal routes, for instance, are considered to be contiguous if the corners of their cells are touching. This will save trace thickness when routing, as you do not have to thicken out the traces artificially in order to have cells with adjacent sides.
- The output of your routing algorithm may be another *.csv file, or it may be displayed in the console as a simple matrix. The choice is yours.
- The examples given in this document are just that: examples. Your solutions do not need to look like these.
- These remarks apply to all of the steps in this challenge.

[illegible]

A possible solution for this step would look like this:



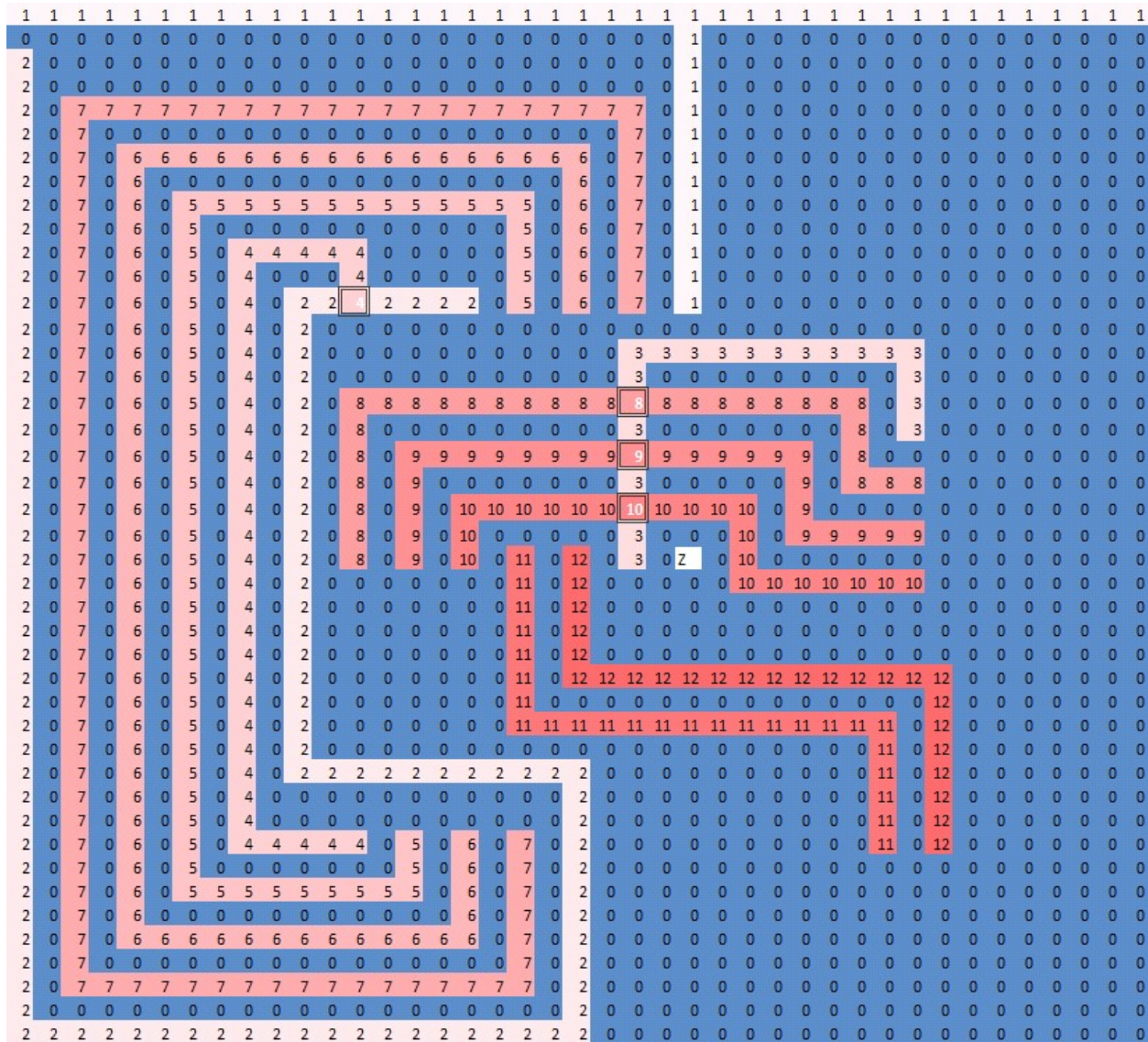
As you can tell, routing difficulty has increased.

Step Three

This step is all about continuity testing.

The package also contains example solutions in *.csv format ("Step_One_Solved.csv", "Step_Two_Solved.csv"), as well as the input for this step, "Step_Three.csv".

"Step_Three.csv" is poorly routed. Some traces intersect, and it's up to your software to say which ones.



You may choose to display the offending routes in the console, in an output *.csv file or you may even highlight them on the UI, if you so choose.

If you elect to display them in the console, a recommended format would be:

Intersecting routes:

-Route One: pins 3, 8, 9, 10

-Route Two: pins 2, 4



If you want to output the result to *.csv file, replace all of the cell numbers (including the start and end pins) on all of the intersecting routes with the “*” character.

When you’re happy with the performance of your continuity testing algorithm, you may even use it as a double-check method to verify your solutions for the first two steps.

Step Four

A user interface. The design is up to you.

Bonus Round:

An interface which allows the user to draw his own pins on a custom grid and submit them for routing.

Good luck!