

Programming Assignment 4

Darin Goldstein

1 Deadline

4/26/2021 at 5 PM

2 The Maze

You are in charge of setting up the scene for a remake of the Shining, and the director is thinking about how he wants the hedge maze in the end scene to look. The hedge company, in order to make sure that it allows for the most general possible maze, will deliver a large $n \times n$ grid. The director will then walk through the grid and rate each wall of the grid, the higher the rating, the more he'd like to knock down the wall.

Unfortunately, the director is not a computer scientist and does not take into account the fact that the end result needs to be a maze; he simply rates the walls only taking into account how nonphotogenic they are. The director is counting on you to create the hedge maze.

1. The outer edges of the grid must be kept intact. You cannot knock down an outer wall.
2. The path structure created after walls are knocked down must be a tree.
3. The maze must be fully connected. From any point in the grid, you must be able to reach (without walking through any walls) any other point in the grid.
4. Of all the mazes that satisfy the above constraints, your maze must knock down the largest possible one (as measured by the *product* of the ratings of the walls you knock down).

3 Input/Output

The maze will be given to you as a sequence of edges followed by their ratings. Because the edges are from a grid, it will be easy to list the ratings. The director will first concentrate on the vertical edges. First, the director will list all the vertical hedges in the first row from left to right, then the vertical edges in the second row, and so on. Once the director is done with the vertical edges, he will

do the same for the horizontal edges. The director will list all the horizontal edges in the first column from top to bottom, then the horizontal edges in the second column, and so on. *No outer edges will be listed.*

You will output a list of the hedges that you intend to prune away by specifying the grid squares on either side of the hedge. (0,0) represents the top-left square of the grid. (1,0) will be the grid square directly below and (0,1) will be the grid square directly to the right.

4 Example

20808 83585	44010 42355	 55933
45310 81762	55574 95417	 67484
79640	79941	

For example, a simple 3×3 grid would have input (input.txt) that looks like the following:

```
3
20808.23681862669
44010.470061730164
45310.41117747684
55574.05247071706
79640.72822573193
79941.74421540713
83585.09822361538
81762.74242870876
42355.44434229584
95417.12587896537
55933.509835449615
67484.88062726776
```

The output should look like this:

1 0 0 0
2 0 1 0
2 1 2 0
1 1 2 1
2 2 2 1
1 2 2 2
0 2 1 2
0 1 0 2