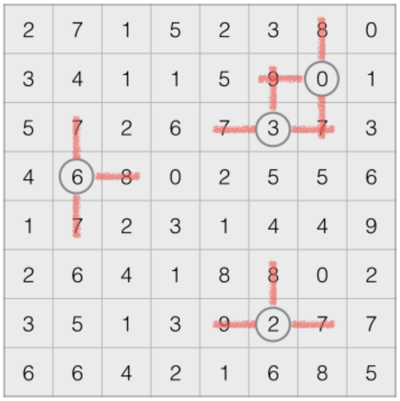
Computational Problem Solving CSCI-603
Lasers! Lab 5

9/27/2015

1 Problem

In this puzzle, you are given a grid of numbers, and a set of three-way lasers to place. Each laser is placed on one square of the grid and covers three of the four horizontally or vertically adjacent squares. Your goal is to cover the highest sum of numbers possible. Note that if two lasers shoot at the same square, you can count that value twice, but you cannot have two lasers centered on the same square. You also may not place a laser such that it would shoot outside the grid (this would be very dangerous to spectators!). For example:

2	7	1	5	2	3	8	0
3	4	1	1	5	9	0	1
5	7	2	6	7	3	7	3
4	6	8	0	2	5	5	6
1	7	2	3	1	4	4	9
2	6	4	1	8	8	0	2
3	5	1	3	9	2	7	7
6	6	4	2	1	6	8	5



2	7	1	5	2	3	8	0
3	4	1	1	5	9	0	1
5	7	2	6	7	3	7	3
4	6	8	0	2	5	5	6
1	7	2	3	1	4	4	9
2	6	4	1	8	8	0	2
3	5	1	3	9	2	7	7
6	6	4	2	1	6	8	5

For the grid on the left, the optimal solution is shown on the right, with lasers that score (from the top) 24, 23, 22, and 24. Note that a laser placed in the same location as the bottom-most one in the solution, covering the squares to the left, above and below would score 23, but this is not allowed (we already have a laser there that scores 24).

Implementation

Implementation will be done in teams of two (you may work alone with prior permission of your instructor). You do not have to work with another member of your problem-solving group.

You may find it useful to sort some items in your solution. To do this, please use and modify one of the sorts presented in lecture this week. Leave this in a separate file, making sure to include the proper authorship tags (but if you modify it, you should add yourselves as authors). You can then import this into the file with your main code.

The grid for the puzzle will be given to you in a file, simply as integers separated by spaces. For example, the small puzzle from problem solving would be in a file with the following contents:

```
3 4 7 9
2 5 1 4
3 3 2 3
5 6 8 7
```

Your code should prompt for a file name, read in the grid data, and then prompt for the number of lasers to place. Your output should give the positions and orientations for the solution — you may designate the four orientations any way that you wish as long as it is clear. For the large example shown here, if we say that the laser "faces" the center of the three squares being targeted, the output would look something like:

```
(6,1) facing west
(5,6) facing north
(5,2) facing north
(1,3) facing east
```

where the coordinates are (x,y) counting down form the top left with (0,0) the upper-left-most square.

Submit your code via the `try` system — `try` will expect one file named `lasers.py` but will also accept any other `.py` files:

```
try grd-603 lab5-1 lasers.py your-sorting-module.py
```

Survey

As part of a research project to determine the effectiveness of different ways of teaching computer science, we would like to get your feedback on this lab. All data will be held anonymously and in the aggregate, only the fact that you have filled out the survey will be used toward your grade. Please go to MyCourses and find the survey entitled "Lab 5 feedback" in the "Surveys" section of the course, and complete the survey by midnight, one day **after** the lab itself is due. Thanks!

Grading

Your grade will be determined as follows:

- 20% : problem solving participation and solutions
- 30% : proper generation of all potential laser placements
- 30% : proper selection of optimal laser placements, including avoiding placements at identical locations
- 5% : proper file handling
- 10% : proper documentation and coding style
- 5% : completion of survey