Stephen Crook

GEOG 683

9-25-14

Lab #4: Options A and D

Lab Objective:

I did two of the options, A and D. For A, the objective was to accept user input for a number of points, and calculate the total distance between a path connecting them, in sequence.

For D, the objective was to prioritize a hypothetical building site based on price and attractiveness. User weights for the importance of both price and attractiveness are input, and results differ depending on these weights. Ultimately, the top two pixels are selected as the building site.

Approach:

For A – I asked the user how many points they wished to enter, and created lists for the X and Y values they entered. Then I modified a previously written function for distance between points to take two x values and two y values as arguments. Then the distance function is applied to each pair of points in order and added to a total distance variable.

For D – Rasters are input without their headers using the raster module shown in lecture. Then the two rasters for price and attractiveness are displayed. User input for weight values is obtained. Using parts of the equation in the lab specs, weight values are applied to each value in both the price and attractiveness rasters, resulting in new rasters of weighted values. Using numpy, the two rasters are added together, resulting in a raster with total utility values for each cell. Then, the highest value is identified. The raster is searched for a second instance of this highest value. If it is not found, the second highest value is identified. Numpy.where() is used to identify coordinates where the highest and second highest values are found, outputting them into an array of x values and an array of y values. These arrays are deconstructed into separate X and Y lists, which are more easily used to reformat into the desired output. Then, results are displayed by either showing the pixels that share the highest value, or the highest value and the cells that either have the second highest value or share it.

Topics Learned:

I learned about using externally built modules in my own code (the raster module). I learned how to approach a larger problem with many extra steps, solving individual problems along the way.

Conclusion

It was a good challenge to have a problem that resembled a real world problem. I really hope it works on all possible rasters (especially the test raster!) There are a number of things I could have improved upon, but having generally gotten the desired results have chosen not to continue figuring out. These include giving an error message if the input weight values are outside of the specified range. Also, a couple extra if/else statements could make the output cleaner (ie, differentiate between cases when there are two of the highest value and more than two of the highest value – right now the output is just “choose these two (or two from this list)”.

**Enclosures**

attractiveness.txt

Lab4\_Report\_SC

Pointspath.py 🡨----- Part A

price.txt

raster.py

raster.pyc

taskD.py 🡨----- Part D