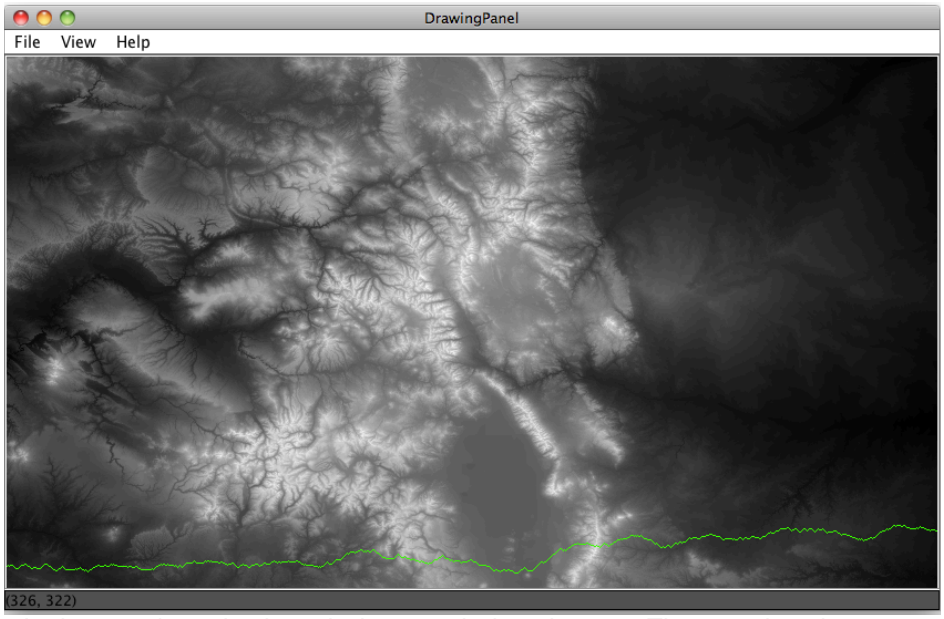
Programming Assignment: Mountain Paths

In this lab you will read a set of topographic (land elevation) data into a 2D array and write some methods to compute some paths through the mountains as well as visualize them.

Background:
There are many contexts in which you want to know the most efficient way to travel over land. When traveling through mountains (let's say you're walking) perhaps you want to take the route that requires the least total change in elevation with each step you take – call it the path of least resistance. Given some topographic data it should be possible to calculate a "greedy lowest-elevation-change walk" from one side of a map to the other.



In the map above, brighter shades mean higher elevation. The green line shows a lowest-elevation-change route for this map, traveling west-to-east.

A Greedy Walk

A "greedy" algorithm is one in which, in the face of too many possible choices, you make a choice that seems best at that moment. For the maps we are dealing with there are 7.24×10^{405} possible paths you could take starting from western side of the map and taking one step forward until you reach the eastern side.

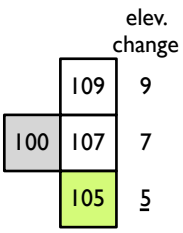
3011	2900	2852	2808	2791	2818
2972	2937	2886	2860	2830	2748
2937	2959	2913	2864	2791	2742
2999	2888	2986	2910	2821	2754
2909	2816	2893	2997	2962	2798

Shows a portion of the data.
Greedy path shown in green.

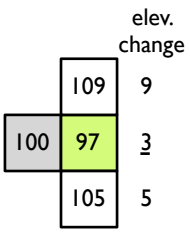
Since our map is in a 2D grid, we can envision a "walk" as starting in some in some cell at the left-most edge of the map (column 0) and proceeding forward by taking a "step" into one of the 3 adjacent cells in the next column over (column 1). Our "greedy walk" will assume that you will choose the cell whose elevation is closest to the elevation of the cell you're standing in. (NOTE: this might mean walking uphill or downhill).

The diagrams below show a few scenarios for choosing where to take the next step. In the case of a tie with the forward position, you should always choose to go straight forward. In the case of a tie between the two non-forward locations, you should flip a

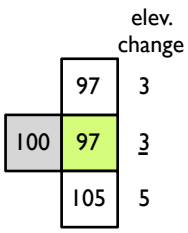
coin to choose where to go.



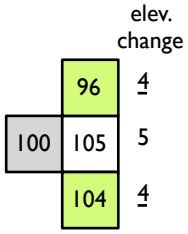
Case 1: smallest change is 5, go fwd-down



Case 2: smallest change is 3, go fwd



Case 3: smallest change is a tie (3), fwd is an option, so go fwd



Case 4: smallest change is a tie (4), choose randomly between fwd-up or fwd-down

There are other ways to choose a path through the mountains that can be explored in the "above and beyond" section of this assignment.

Assignment Requirements

The minimum requirements for the assignment are that you write code to produce something like the map shown in the picture above. To do that you need to complete the following 5 steps:

Step 1: Read the data into a 2D array
Step 2: Find min, max elevations, and other calculations on the data
Step 3: Draw the map
Step 4: Draw a lowest-elevation-change route starting from some row
Step 5: Find and draw the lowest-elevation-change route in the map