

CSCE 206 Final Exam Take Home Portion

Due 9:30am Thursday December 13, 2012

Submit via dropbox

If we ignore plate tectonics, we can model changes in the surface of the Earth as becoming flatter due to wind and water erosion. Given a terrain dataset from the present, we should be able simulate this flattening process over time. Assume that terrain data is read from a file and stored in a two dimensional array. The first line of the data file also provides an integer value for the number of rows and an integer value for the number of columns of the area. The elevation values in subsequent lines are of type double. Given the current grid position `grid[m][n]`, after one million years, the altitude of this grid position could be projected to be the average of all the adjacent grid positions including itself. In other words, the new grid value will become:

$$\text{new_grid}[m][n] = (\text{grid}[m-1][n-1] + \text{grid}[m-1][n] + \text{grid}[m-1][n+1] \\ + \text{grid}[m][n-1] + \text{grid}[m][n] + \text{grid}[m][n+1] \\ + \text{grid}[m+1][n-1] + \text{grid}[m+1][n] + \text{grid}[m+1][n+1]) / 9$$

However, due to inhomogeneities in wind and water erosion, the altitude of the new grid may fluctuate by as much as 20%, which means

$$0.8 * \text{new_grid}[m][n] \leq \text{new_grid_real}[m][n] \leq 1.2 * \text{new_grid}[m][n]$$

The position `new_grid_real[m][n]` should be uniformly distributed between the upper and lower bound. Hint: This effect could be simulated by a random function.

Write a program to read in a grid data set, simulate the flattening process and generate a file containing the new grid. The input to your program should be a file called "CurrentSurface.dat". The output file should be named "FutureSurface.dat". You may assume that the grid will not exceed 500 rows by 500 columns.

Tip1: If a grid element is on the outside border (in other words it does not have neighbors on all sides), then you will only need to consider those neighboring grid elements with valid indices.

For example consider the element `grid[0][1]`:

$$\text{new_grid}[0][1] = (\text{grid}[-1][0] + \text{grid}[-1][1] + \text{grid}[-1][2] \\ + \text{grid}[0][0] + \text{grid}[0][1] + \text{grid}[0][2] \\ + \text{grid}[1][0] + \text{grid}[1][1] + \text{grid}[1][2]) / 9$$

However there is no row -1. Consequently, this grid element has only 5 neighbors. So the computation to compute the new element includes those 5 neighbors and the element itself:

$$\text{new_grid}[0][1] = (\text{grid}[0][0] + \text{grid}[0][1] + \text{grid}[0][2] \\ + \text{grid}[1][0] + \text{grid}[1][1] + \text{grid}[1][2]) / 6$$

Notice that the total number of elements in the computation is now 6 instead of 9.

You might consider using if-statements to handle out-of-range indices. Keep in mind that since the array describes a rectangle, there are border elements on all four sides.

Tip2: It would be a good idea to read your textbook's approach to terrain navigation starting on page 398. While the problem in section 8.2 is not identical to your problem, the textbook discussion could help you to get better understanding of the problem and help you in finding a solution.