ITCS 3112 — Project 3: Mountain Paths Using Pointer Representation — Fall 2017

Due Sept 27, 11.59pm

This project is a continuation and modification of project 2, differing in the following aspects:

1. You will use a **pointer** representation to access the data
2. You will allow the user to accept **multiple starting points** and display all the paths in your output image.

Data Representation.

Instead of using a static 1 dimensional array representation (as in project 2), you will use a **dynamic array** representation. You will start with an array of (row) pointers that point to each row of pixels in the image. Each such pointer is itself a pointer to an array. This can be done as follows:

*int **elev_data = new int* [imageHeight];*

This call creates a pointer to each row of pixels (type is int*, and assumes each pixel is of type int) in the image, with elev_data[k] pointing to row k. What is left is to allocate an integer array (of size imageWidth). This can be done in a loop for each row of the array above.

*row[k] = (int *) new int[row_width];*

The resulting elevation dataset can still be accessed using 2D array indexing, since the first index points to the row and the second index points to the column.

Once the data has been used and done with, it should be deallocated, using the **delete** method. For an array its of the form

delete [] arr;

where arr represents the array to be freed. In our case, you must free each row of the data, followed by freeing the array of row pointers. Note that you will deallocate in the **reverse** order of the allocation.

Datasets: Use the two datasets from project 2.

Tasks.

1. **Read dataset, create image.** Similar to project 2, but using the new representation for the data. Output the original data in PPM, then convert to PNG.
2. **Compute Multiple Elevation Paths.** You will take a set of row numbers from the user (do at least 3), run the greedy algorithm on the image, output paths to the image.
3. Make sure you free all allocated data prior to terminating the program.

Requirements, Evaluation (See Rubric on Canvas):

- You can implement this project as a purely procedural program, similar to project 2.
- You must use the data representation as specified above, using new and delete to manage memory.
- **Input/Output.** Your program will take as input a set of row numbers (at least 3, column is assumed to be 0) and an input file name; and it will output the original image and the image with all elevation paths drawn. **Run your program on both datasets and show the paths generated; turn in the output images in PNG format**.
- **To Turn in to Canvas: Your source code files, two images in PNG format showing elevation paths, for the two datasets. Include instructions on how to run your program. Source code must be well documented for full credit.**