**MCS 3312 Analysis of Algorithms – Fall 2017**

Homework Assignment #2

Jack Baumann

**Background:** Hilbert curves (<https://en.wikipedia.org/wiki/Hilbert_curve>) are rapidly becoming a major player in object recognition from imagery. In this homework assignment you will practice the methods learned in class as well as gain exposure to some basic concepts from modern image processing.

Problem 1: Homework2Problem1.py

Use a decision diagram to compute the Hilbert curve for a space up to a depth *d*.

Part A:

Write a program that

1. Queries the user for an arbitrary depth *d*.
2. Stores a copy of the progenitor for the Hilbert curve (three line segments).
3. Is capable of constructing a decision tree with a branching factor of 4, and navigating this decision tree to the input depth *d*.
4. Render the entire Hilbert curve correctly to the screen using Panda3D. (Use the decision tree and dynamic programming to help define the curve).

Part B:

Compute the computational cost for your method (show your work).

Total computational cost: d(12n) + 2n + c

Problem 2: Homework2Problem2.py

Compute an approximate Gaussian pyramid (<http://www.cs.utah.edu/~arul/report/node12.html>, <https://en.wikipedia.org/wiki/Pyramid_(image_processing)#Gaussian_pyramid>) for an input file.

Part A: homework2.csv

Manually define a comma-separated-text file and populated with random values. (Do not submit your comma-separated-file. I will define my own file for testing.) Write a program that is capable of correctly parsing the text file and write the output to the command prompt using an appropriate, easily readable format. The file should have the following format:

2,3,4,3

4,5,5,9

5,7,3.4

3.6.8.2

and the input in the file should be both square and a power of 2 in terms of rows and columns.

Part B:

Use dynamic programming to compute the approximate Gaussian pyramid for the input text file.

1. For the input *n* × *n* input matrix, compute a new *n*/2 × *n*/2 matrix that averages each 2 × 2 cell from the original matrix into a new cell in the *n*/2 × *n*/2 matrix. You should have log *n* such matrices/images.
2. Print all log *n* matrices (original, intermediary, and final) to command prompt using an appropriate format.

Part C:

Compute the computational cost for your method (show your work).

Total computational cost: (1/2)n^2 + cn + c

Problem 3: Homework2Problem3.py

Use the program from 1A and the program from 2B to generate a Hilbert Curve with dynamic depth.

Part A:

Write a program that

1. Computes the approximate Gaussian pyramid from an input comma-separated-text file. homework2.csv
2. Utilizes the averaged values as decision factors that determine depth for the decision tree in the method from 1A. The way that this should work is that each position in your decision tree corresponds to a position in the approximate Gaussian pyramid. So, the value at that position is used to decide if the Hilbert curve should continue down the decision tree. The check that needs to be performed is whether or not the average weight of the next level nodes is higher than that of the value for the current position in the current image. If a child node has a higher value, then continue on down the decision tree. If it does not, then do not visit that child node. (Don’t forget to render your derived Hilbert curve to the screen using Panda3D.)

Part B:

Compute the computational cost for your method (show your work).

Total computational cost: (1/2)n^2 + 25n + 4log(n) + d + c

Problem 4: Homework2Problem4.py

Use the program from 3A to convert a grey-scale bitmap into a Hilbert curve.

Part A:

Write a program that does the same as in 3, but that uses a bitmap in place of a comma separated file.

Part B:

Compute the computational cost for your method (show your work).

Total computational cost: (3/2)n^2 + 25n + 4log(n) + d + c

Problem 5: (bonus)

Part A:

How can a Hilbert curve be utilized to uniquely identify an object in an image?

Hilbert curves can be used to compress images and find patterns in them. An image of a cat will have sharp differences in curve depth around its ears, whereas a shoe will be smoother and more uniform. By utilizing Hilbert curves, images of different objects can be uniquely identified by the patterns in them.

Part B:

If it is needed, then why would a Hilbert curve be needed to uniquely identify an object in an image?

A Hilbert curve would be needed to uniquely identify an object in an image over another method because of the simplicity of the fractal it creates. It is easy for a human or computer to identify an object from the Hilbert curve by looking at the recurring patterns and differing depths found within the curve. Machine learning can compare patterns found in a given Hilbert curve to Hilbert curves of previously identified objects to identify which object the given curve best represents.

To turn in your homework, submit a separate word document for the short answer portion of the homework (do not include the short answers in your code.) For the code, submit a separate Visual Studio workspace for each problem (containing folder all). Before submitting, make sure that the code runs, and that it is not dependent upon any libraries that are not in the supplied project. At least one of your computational cost estimates should utilize the master’s theorem.