**Allen Spain**

**CSCI 6810**

**June 06 2020**

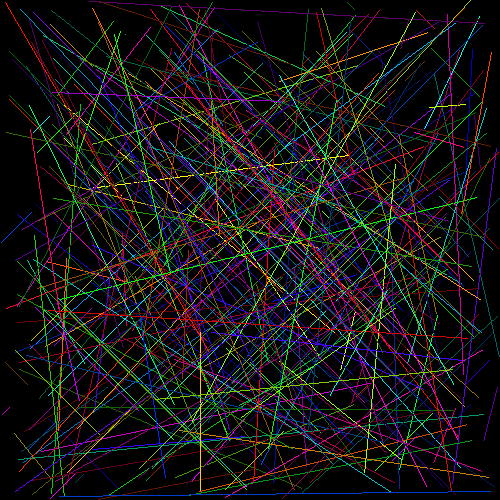
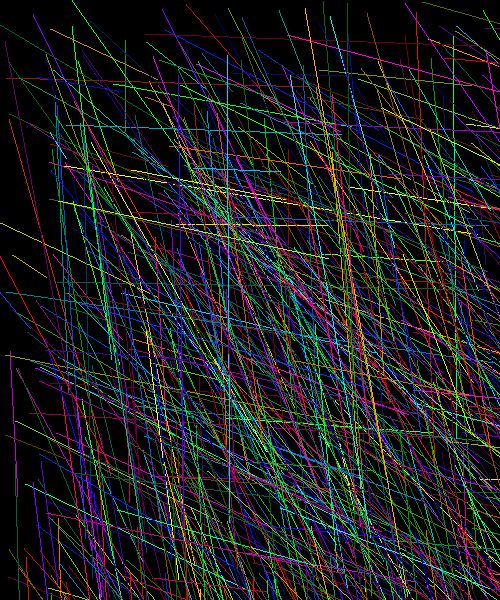
**P1-2**

**# 1** see video submission

**# 2**

Basic alg & Bresenham can handle all of the below examples (see video)

Basic Algorithm Bresenham algorithm



All of the following also work for both algorithms

# a: (10,10,10,30)

# b: (30,10,10,10)

# c: (10,30,10,10)

# d: (10,10,20,30)

# e: (10,30,20,10)

# f: (20,30,10,10)

# g: (20,10,10,30)

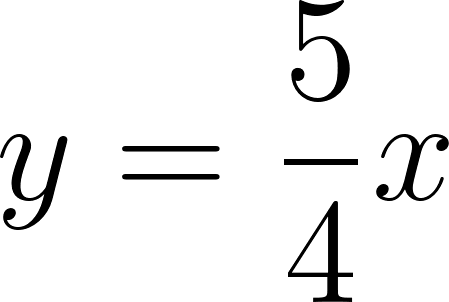
**# 2 & 3:**

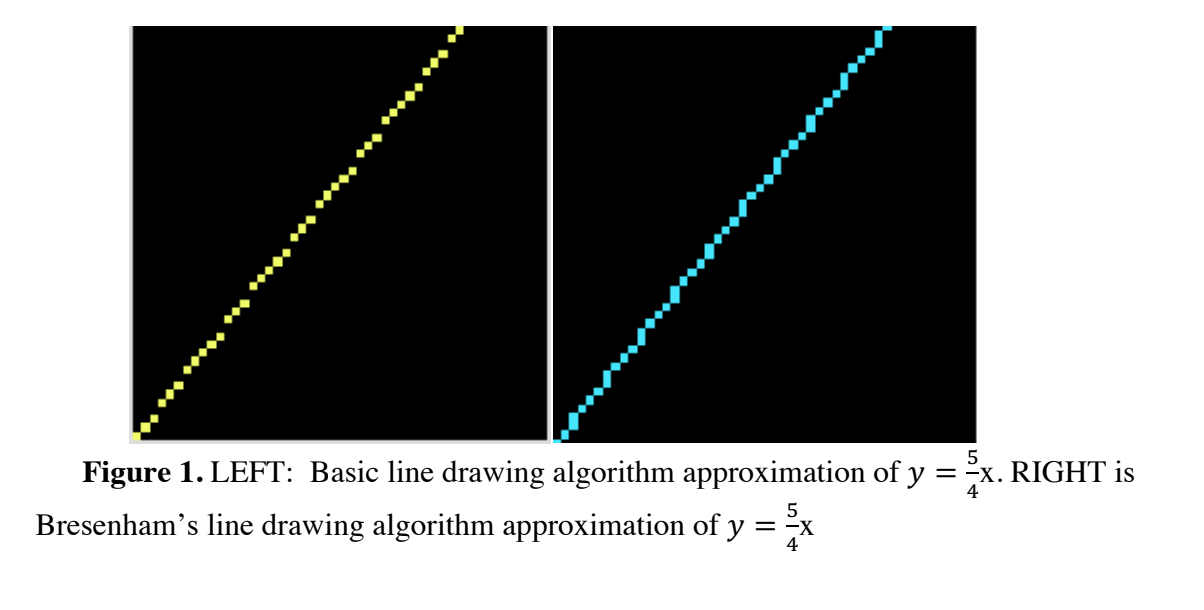
**I. Experiments**

Both implementations of the Basic and Bresenham’s algorithm exhibit the capability of being able to fill contours. The geometric shapes tested are: the rectangle, square, and isosceles triangle. This functionality can be demonstrated during the demonstration.

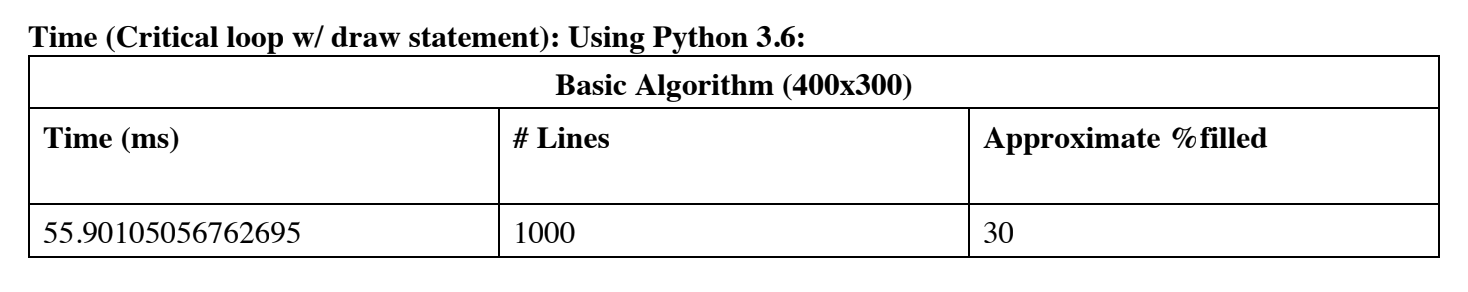
**II. Efficiency**

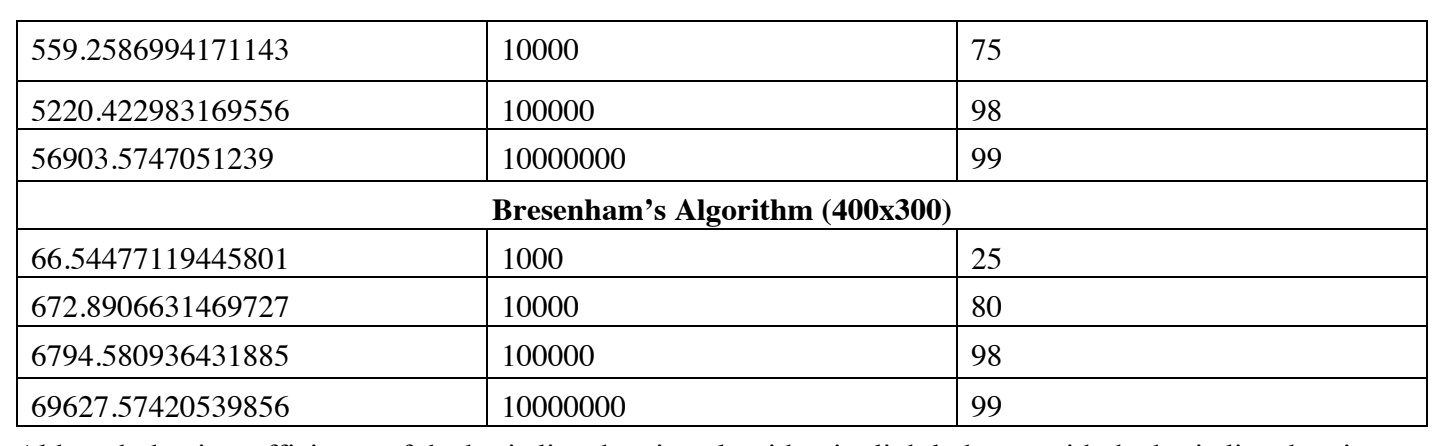
**Accuracy (line appearance):**

The simple function of [](https://www.codecogs.com/eqnedit.php?latex=%20y%20%3D%20%5Cdfrac%7B5%7D%7B4%7Dx%20#0) was used in order to demonstrate the difference in line-drawing efficiency across the two algorithms examined in this assignment, observable in fig 1. It is apparent that the Bresenham’s integer approximation approach is much more accurate, in that its appearance has fewer gaps, and matches the slope better than its opponent. Although the difference or “error’ function implemented in the Bresenham’s adds complexity, the result is a better approximation in a discrete environment than simply using slope intercept form and round each successive y value for its corresponding x as described in the slope intercept form [](https://www.codecogs.com/eqnedit.php?latex=y%20%3D%20mx%20%2B%20y_%7B0%7D#0)



Time complexity:





Although the time efficiency of the basic line drawing algorithm is slightly better with the basic line drawing algorithm, Bresenham's line drawing algorithm is still a preferred line drawing method due to the rationale described in the Accuracy section.