Joshua Ungheanu & Raul Garcia

Project 1 Section 335-05

**Complexity of our Langton’s Ant problem.**

The initial configuration of the program is composed of all black cells at state 0 with random conditions of colors red, blue, and yellow at every turn. Our ant behaves in the manner at which it changes its current black cell state to red (1), then turns 90 degrees to the right and changes color to yellow (2) and then moves one cell forward. During the process, the ant runs into cells that have already been visited, during the first few iterations out of 1000 the ant begins to reveal a somewhat symmetrical pattern. The ant will return periodically to the central square from which it started.

Big O running time of the program

The running time for our Ant Cella algorithm is 0(N). The function run recursively n times since any function running inside the move function with be nested.

Main Operations

The main operations is the move function that passes seven parameters which initializes the movement of the arrow(ant). Depending on those factors, the ant should end up designing a specific pattern since the movement has been set to move 1000 steps.

Styles: Running time of 0(1).

drawTriangle: Will run without any loops, or recursion. Running time of 0(1).

drawGrid: Two loops for height and width. Running time of 0(n).

drawSquare: Will run without any loops, or recursion. Running time of 0(1).

move: Boolean that checks for the state. Running time of 0(1).

setTimeout: Will run without any loops, or recursion. Running time of 0(1).

Final Running time of 0(n).