David Nguyen

Professor Chuck Siska

CPSC 335-05

28 September 2019

Analysis of Big O

The helper functions draw_triangle() and draw_rect() have constant O(1) comparison operations. The draw_grid() function has two linear O(N) modification operations that draw lines on the x and y-axis. The main focus of the algorithm analysis is on cella_ant() function. The cella_ant() function is an LLRR traversal algorithm that is responsible for traversing the grid, incrementing a table element, and changing the color according to its current state. The input size of the algorithm can be as large as a 41x41 grid and cannot exceed the bounds of the grid. That means the algorithm can have more than 1681 steps because each element in a 41x41 grid has four states but cannot exceed the upper limit of the state evaluation loop. Also, if the current element state is greater than four, then the state can be reset back to the zero. The traversal algorithm can run until the upper limit of the state evaluation loop. The main operation of this algorithm is not the initialization of the lookup table because the table is not the main focus of the traversal algorithm. The main operation is the loop that evaluates the state of the current element on the lookup table. After temporarily utilizing the console count function for each operation, the evaluation loop is responsible for calling these follow functions up till the upper limit of the loop: draw_rect(), draw_triangle(), state increments. The evaluation condition in the loop is constant O(1), but it does not matter so much if we are trying to analyze the Big O Notation of the algorithm. The Big O Notation of the main loop that evaluates the current states

is linear O(N). Even though there is a quadratic O() nested loop in the cella_ant() function that initializes the lookup table, this is not the main focus of the algorithm.