Complexity Order

The main function used to show the generational progress of Cella Rule-45 is Cella45Gen. Using an array to represent each cell, Cella45Gen checks each cell and determines the next state based on Cella rule 45. After determining the next state, the function then changes the state in the array and then displays the change through the html canvas. Cella45Gen uses brute force as it checks each cell and applies a set of rules to it. The big-o of Cella45Gen is O(n2) with input being the number of cells within the square grid. Cella45Gen is O(n2) because this function uses a nested loop with both loops having a running time of O(n), directly proportional to the given size of the square. Since it is a nested loop, both having a running time of O(n), Cella45Gen has a quadratic complexity based on the number of cells within the square grid. All other functions and operations are at most O(n2) running time as they have a similar structure to Cella45Gen, specifically using nested loops.

Cella45Gen: (Not exact code)

function Cella45Gen(rctx,array, columns, rows){

for(var y=1;y<rows;y++) (O(n))

{

for(var x=1;x<columns+1;x++) (O(n))

{ array[y][x]=nextState(array[y-1][x-1],array[y- 1][x+1],array[y1][x]);

if(array[y][x]==1)

{fillcell(rctx,y,x);}

}

}

}