Rachel Pierstorff

Project 3

April 21, 2017

The differences between the depth and breadth search algorithms arise from the data structures they use. The depth algorithm employs a stack with a last-in, last-out system. As it goes through the grid, it moves vertically and adds neighboring vegetation cells to the cellsToExplore stack as it finds them. Because of the LIFO system, the most recent cells added (the vegetation surrounding the current cell), are the next cells explored. This means that the algorithm moves as far down one path as it can until it reaches the bottom of the grid or a cell no longer has neighbors.

The breadth search, on the other hand, is a queue with a first-in, first-out system. This means that the cells from the first row are explored first, their surrounding vegetation noted—but added to the end of the list. Once all the neighboring vegetation cells are added to the list, the next cell is explored, checked to see if it is in the bottom row, and neighbors added again. This search algorithm is significantly less efficient than the depth search; because every cell on the list is explored in the order it was found, the algorithm searches the list horizontally (that is, across rows rather than down columns)—jumping from possible path to possible path, while the depth search has an increased likelihood of finding a successful path on its first or second try.

Essentially, the breadth search spreads fire horizontally, while the depth search spreads fire vertically; because the path is deemed successful based on its ability to transverse the forest vertically, the depth search is more efficient.