The improvements that were decided on and implemented into the program were creating a sudoku solver for a 3D puzzle. The solver works only with specified layer values, for testing, the number of layers was set to 3. To check if a number could be inserted in a certain cell, we used edges. The edges between certain vertices would represent their relationship, meaning that if two vertices were connected by an edge, that meant that they were neighbors. This meant that checking if a number was valid would-be O(n), n being the number of edges the vertex had.

The question of if it’s better to find many solutions or finding the faster solution all depends on the context. If it’s unknown whether a solution even exists, then the BFS approach may be the better choice. BFS, although slow, will find any and every solution to a sudoku puzzle. If the solution counter from the BFS ends up being zero, then no solution exists. However, if the solution counter, for a sudoku solving algorithm using a DLS approach, ends up being zero, that does not necessarily mean a solution does not exist. DLS may not be the best approach for finding solutions, but in terms of reducing time and space complexity it is certainly the better choice. Overall, whether one should use the DLS or the BFS solution approach depends completely on the context. If one wanted to optimize time and space in place of finding every possible solution, then use DLS. If finding every solution is more important than saving on time and memory, then use BFS.