Final AI Report

I. Describe what you did to make your Final AI agent “smart.”

For our minimal AI, the first algorithm I used was forward checking. It uses a list of values that each block can possibly have based on other values that are assigned. For example, if one block is assigned to 5, the other blocks that are at the same row and column cannot be 5 on the Sudoku board. Therefore, all the possible values for each neighbor of a block will change once the value on that block is assigned. In the code, I first created a list containing all the variables that is assigned and do the algorithm for each variable in the list. In draft AI, I implemented MRV and LCV algorithm. MRV as known as minimum remaining values, is an ordering strategy in backtracking search. I use the block that has the fewest remaining values in its domain to determine which block we assign next. LCV is the term for least constraining value, it rules out the fewest possible solutions below its node, and we will have the most chances to find the solutions early Sudoku board. According to the description, I sorted the list we have after algorithm and return it as correct format. In final AI, we implemented NOR and MAD. In norvig check, I followed the pseudo code from the slides to eliminate the value from the square’s neighbors if that the variable is assigned, and if a constraint has only one possible place for a value, we put the value there. In norvig check, I used both constraint propagation and check the consistency of the network to make sure it runs properly as it makes our final AI agent smart. The last one we implemented was MAD, which is the term for MRV with tiebreaker. This algorithm is similar to minimum remaining values, we will need to return the unassigned variable with the smallest domain and affected the most unassigned neighbors. In this algorithm, we need to take account the situation when there are multiple variables that have the same smallest domain with the same number of unassigned neighbors. If this is the circumstance, we will need to add them to the list of variables. In addition, if there is only one variable, we return the list of size 1 that contains that one variable. As the result, the return value would always be a list that has one variable.

II. Provide suggestions for improving the intelligence/performance of your agent.

With the current implementation of the solving algorithms, we can further investigate how the different combinations of consistency checks, next variable selection, and next value selection affect the efficiency and performance of the Sudoku solver. We may test each combination of the algorithms against a large set of random boards and calculate their average number of boards solved successfully and backtracks, then compare the results and select the one with high completeness rate and fewer backtracks. Furthermore, we may train the agent to identify patterns in the board and apply the algorithm that has better performance on this board. One parameter might be the to what extent the given values are clustered in one constraint. Although further investigation is required for this improvement strategy, it would make the agent more intelligent by mimicking the Sudoku solving strategy selection process from an experienced player.