Andrew Jacober, Sarah Hand, and Mira Kasari

1. How did you implement your best backtracking search?

We start with a board that has all values of 0, and no assignments yet. We iterate through the board, and if the node is equal to 0, we expand that node and assign it a value in the domain, that only includes values not already in the row and column. Once we fill each node with an assignment, we check the board, which checks columns, rows, and cages. If it returns false, then we recurse again until our checkBoard method returns true.

2. What did you use for a utility function to guide your local search?

We used random-restart hill climbing to guide our local search. We start with a board that does not have any row/column constraints, then we swap the most constrained node with a random node that has at least one constraint. If the total number of problems is now equal to 0, then we have found the solution. Otherwise, if the total number of problems the board has does not change (if the next number of problems is equal to the current number of problems, we are not getting any better, so we are at a LOCAL MAX so we should break and try again), we create a new board with different *random* assignments this time and start the local search again. We continue this until the number of problems is 0, i.e. we have found a solution.

3. How did you choose which nodes to change for each iteration?

For each iteration, we choose the most constrained node, then we swap it with a random node *in the most constrained node's row* that has at least one constraint. To find the most constrained node, we check each node's row, column, and cage for "problems" (if there are duplicates in the row/column or if the cage isn't getting the right solution). Then we create a parallel board that holds the amount of constraints each node has. The most constrained node is the node in the board with the max number of constraints.

3. All group reports need to include a brief statement of individual contribution, i.e., which group member(s) was/were responsible for which parts of the solution and submitted material.

We worked together on all parts of the assignment by physically meeting frequently for the past three weeks. Andrew did a lot of the heavy lifting when it came to figuring out code, Sarah and Mira did a lot of the understanding and interpreting of the desired searching algorithms. But all members coded and researched heavily.