N number of Trials	Hill Climbing Search			
	Successful Runs	Total Runtime(ns)	Average Runtime(ns)	Average Steps
100	0	122143104	1.779637391899E10	4.32
500	1	251491728	3.406157325048E9	5.004
1000	7	410723071	324395804364E9	3.458
2000	1	673742906	9.07541416374E8	4.8865

N number of Trials	Min-Conflict Search			
	Successful Runs	Total Runtime(ns)	Average Runtime(ns)	Average Steps
100	12	4076646959	1.446141621216E10	880.37
500	65	19974836150	3.406660567916E9	870.39
1000	150	39596651803	324807676439E9	850.42
2000	286	79409020296	9.07878347987E8	857.414

This project was made in order to solve an NxN queen problem, but for this specific project, it was specified that N = 14. I approached this project with a view of effectively utilizing two other classes representing various objects in the queen problem, those being Queen and Board, with their own respective properties and functions to assist in solving this problem.

From these results, the project had a pretty bleak outlook. Hill Climbing had drastically low success rate almost all the time. If anything, even one or two successful runs have mostly to do with luck. Although, keeping in mind that steepest hill climbing search can only solve approximately 14% of the puzzles given at an 8x8 board, it makes looking at these hill climbing search results a little bit better, but still quite depressing. My take on this search algorithm was to continuously return a single board that hopefully had the best heuristic value possible. After, it would compare with the current board, however one thing about my code that I do not like is that I do not have a surefire method to continuously try to achieve a better board. At the moment, it currently stops when the original heuristic is equal to the neighbor’s heuristic in hopes that this Boolean expression would give the algorithm some more attempts to try. But whether it is the search algorithm itself, the method of finding a neighbor or both is quite difficult to get.

For the Min-Conflict search results, they were much better, but not as good as I thought they should have been. Min-Conflict is also a much easier procedure to perform I believe, as it is mostly moving individual queens on a board and hoping for the best. However, across my many tests, the Min-Conflict search averaged about a 15% success rate. However, I do believe that for any Min-Conflict problem, an easily solvable problem would only start with a favorable initial state. That may be one reason why there are so many failures for Min-Conflict; however I would think if that were the only reason, the success rate would be a little higher than 15%.