Tony Diep
u0934661
2/3/2017
2420
Program Document Analysis
Milad Bazzazi
I pledge that the work done here was my own and that I have learned how to write this program, such that I could throw it out and restart and finish it in a timely manner. I am not turning in any work that I cannot understand, describe, or recreate. I further acknowledge that I
contributed substantially to all code handed in and vouch for it's authenticity. (Tony Diep)

1. Create a plot of the cost of running (the number of guesses) the recursive solver.
2. What does this empirical analysis make you think about the likely future runs of the program?
3. Come up with something to measure about the constraint solver and create a plot of the cost of running this solver on the sample Sudoku programs
4. Describe what you measured and why you think this is useful.
5. Discuss the constraint solver. How does it make this project work?

Although I do not have a graph constructed for the Sudoku assignment, I will still discuss about the two methods to solve Sudoku puzzles: the elimination method (the constraint method) and the recursive method. The elimination method focuses on pruning and finding possible values, while the recursive method is an iterative process of trial and error in which for every empty spot, it tries all of the numbers 1 to 9. The empirical analysis would represent that as the difficulty of a Sudoku puzzle increases, the number of guesses for each solver will also increase. Some harder puzzles will even eventually lead to such high number of guesses in which could also imply that the Sudoku puzzle is not solvable.

I spent over 16 hours working on this project, which is certainly more than expected. I was overthinking certain methods on the Sudoku class when in reality the solutions were really simple and efficient. I implemented too much code than necessary because when visiting the professor's office hours, he helped clarified what I was doing that made me do more work than necessary. For the past two assignments, I spent too much time how to implement the code. I also spent too much time feeling intimidated on the complexity of the assignment, so I need to focus on completing what I already know how to do.

The most time consuming part of the programming assignment were implementing the two ways of solving the Sudoku puzzles: 1) Recursive method and 2) Constraint solver (elimination). Although the professor wrote psuedocode on how to implement the two methods, I still struggled on translating the psuedocode to Java code. In reality, the implementation was much straightforward after spending a little over 2 hours figuring out how to implement it. Perhaps I should try working on this assignment in a different place besides the CADE lab because that is where I get the most distracted. This is likely because I know a lot of people in there and I spend a lot of time socializing with friends and classmates than spending more time implementing code onto the assignment. I figured if I can try and work on the assignment alone at a different area, then I can get a lot more done and thus spend less hours. This is likely what happened every time I started the assignment as early as possible having to realize that some friends and classmates and I got together to discuss the assignment. However, we spend more time talking about non-related assignment topics and that is likely that I don't get a lot of code done in a timely manner. Of course, I may have planned on doing assignments the wrong way.

Little bugs came up when running the JUnit Tests and I spent too much time figuring out how to get the test cases to display the right results rather than implementing the empty methods in the Sudoku class. While I managed to set up the JUnit Test headers for each method in the Sudoku class, I did not fully worked on the test cases first before starting on the Sudoku class. I also did not have the preparation to work on the assignment in which I was not concentrating hard enough on the assignment. Thus, I spent too

Tony Diep
u0934661
2/3/2017
2420
Program Document Analysis
Milad Bazzazi
I pledge that the work done here was my own and that I have learned how to write this program, such that I could throw it out and restart and finish it in a timely manner. I am not turning in any work that I cannot understand, describe, or recreate. I further acknowledge that I
contributed substantially to all code handed in and vouch for it's authenticity. (Tony Diep)

much thinking mentally on how to implement the code instead of drawing Sudoku puzzles on pencil and paper and pinpointing how to implement them in Java.

Recursion is basic terms is the mechanism of where a method calls upon itself inside its body. In some cases it is actually efficient than nested for loops but that does play into the complexity depending on the size of “N”, where N is the number of elements. Recursion has two cases: one is called the base case (that is, the case that stops the iterative call of calling itself) and the recursive case (that is, the case that is doing the iterative calls upon itself). This mechanism is a good tool to use in this project because without it, we would have to implement lots of nested loops which can be tedious sometimes. When we really think about implementing the solver for the Sudoku puzzle, there are some conspicuous and not-so-conspicuous patterns that we may or may not see directly. When there are some numerical patterns, we can take advantage of them and thus use recursion to get a more concise way of solving Sudoku puzzles in a timely manner. I believe I have a good understanding of recursion but only when I don’t think too hard about it. If I do, then implementing recursion into the Sudoku solver can be challenging.

The project, at first glance, seemed interesting and exciting, but I realized how much I overthink a lot of the problems. There are several patterns in this assignment I should have noticed but they can be pretty complex in terms of detecting those patterns. If I had the opportunity, I could have implemented the GUI for the Sudoku puzzle. That was what I hoped to do, but I overworked myself in which it didn’t equate to the amount of work I should have completed. But I noticed this was something I continued to do ever since the first assignment and that needs to change.