As a team, we went through the JavaScript portion of our data structure visualizer. We walkthrough each part of the button’s function for each data structure that we implemented in the past 2 weeks to find faults or bugs that were not necessarily part of the overall program. Using the verification section from the lecture about software verification, we as a team walkthrough the coding artifact of our finished project. As a result, we have discovered issues with user verification when a prompt was given from a walkthrough of the code. Each data structure has issues when a user has canceled the input, allowing the function to continue the code with a null input from the user. Specifically, stack, queue, and linked list where primarily feature a prompt to input a value from the user. If the user presses space and enter, it would be an invalid value that is inputted into the function of the 3 data structures. Another invalid input that would be documented would be the null input if the user used the cancel button on the prompt. It would allow the function to continue with the null input and output a box without a return value to give back to the user. The assignment of the fault was given to everybody on the team and if they have discovered the solution to the problem, it would be implemented in all of the data structures which used the prompt response of the user to input a value into the function of the data structure. For the stack, the empty string and null input were fixed by Eric by implementing 2 if statements to catch the input and restart the function for the user so it does not continue as a result. For queue, the empty string and null input, the empty string was caught by an if statement then alerts the user to not input an empty string into the prompt and for the null input, another if the statement was created to just return the user to the data structure main page so it would appear to the user that they have canceled the original input, this was fixed by Duy. Finally, the linked list had the same issues as stack and queue where a user would be able to cancel the prompt to give it a null value and be able to take an empty string to give it to the overall function that would cause faults in the entire program, it was noted by every one of the team and Aidan had fixed these issues by including 2 different if statement that would catch the invalid values given by the user to be able to “reset” the function so it would not continue with the invalid values that would mess up the entire code base from it. Another fault that we have noticed from our code review was that the binary search tree did not draw on the webpage correctly when inserting nodes, it was fixed by Landen when the browser that was running the program was switched from Google Chrome to Firefox. In summary, there was a total of 7 faults that were found during our walkthrough through our codebase, fixing all the faults that were discovered from the walkthrough by everyone on the team.