|  |  |  |
| --- | --- | --- |
| **Bug** | **Cause** | **Steps to avoid reproduction** |
| Careless implementation of bit manipulation routines caused bugs. Specific: - subtracting of 1 after the inversion of the bits meant the subtraction was having the wrong effect. | - Lack of thought before implementation of the methods  - Lack of rereading of the new code to understand the code and data flow.  - Lack of thinking about ways that the code could break. | - Spend more time thinking before implementing any code. No matter how simple.  - Make it a note to reread every line of code before moving on to the next task at hand.  - Write test cases first will aid in defining a clear interface and thinking about the potential inputs that will break the code.  - Step through all of the code at least once in order to understand all of the data flow and what is happening in the code. |
| A hard coded value remained inside of the code causing unexpected behaviour. | - Placing a hard coded value inside of a script and not going back to it on the same day. | - Avoid placing hard coded values in code. - If it must be done, write it down somewhere in order to not forget. Also, attempt to return to it by the end of the day as the more time passes, the less likely you are to remember it. |