Story 1:
This story is about the dumbest bug I both created and fixed. So when I was writing my tic tac toe AI I had an issue where it would recursively run itself about 8x more than it should have and I couldn't for the life of me figure out why. I tried adding console.log statements wherever I could trying to figure out why it was running iteration 8 13 times but to no avail. It turns out, I needed to add another check for if the board was full, which I hadn't done previously as I thought having my function to check for a winner was enough, but apparently it was not. Afterwards it worked like a charm, still took a bit longer than I liked though.

Story 2:
This story is about my adventures learning multithreading in python. This was about a year and a half ago, and I was writing a script to mass-download images from assorted image sites, and I was running into an issue where it would only download 1 image at a time, which made it take a lot longer than it needed to, especially if you were downloading a few thousand images (which I was doing for training a machine learning algorithm). After trying a few things to remedy it, I decided to give learning multithreading a try, and hoo boy was that an experience. The first issue I ran into was rate limiting, since I didn't limit the amount of image download requests my script could make at any given time, so it was trying to download all 250 images at once, which did not make any image hosting site happy. After adding some limiting to that I ran into issue number 2: file locks. So for some reason, despite me not telling it to do this, it tried to save all the images to the same filename and would constantly error itself out since it was trying to write to the same file with multiple threads, which the kernel does *not* like. I had to rewrite a lot of my file-saving function to get that to work, and even then it went back to only downloading one thing at a time, at which point I gave up. Mayhaps one day I will return to it and once again try to understand multithreading.