

## Milestones and Timeline

For this assignment, I was better able to follow my guideline for myself, though not completely, and not yet satisfactorily. I had planned for a few days over the course of the week and a half since the last programming assignment, though not all of the days would be spent coding. I was able to start earlier, not just with planning the project, but also with the coding of preliminary things, like function headers, makefiles, and tests. I planned and coded earlier, but I still was met with the bulk of the work the weekend before it was due. Again, I have had tests every week for the past few weeks that I needed to prioritize, which is why I couldn't do more of the work earlier. I think I handled my time better this time. An hour before it was due, I was able to complete the assignment satisfactorily, with only a few test cases incorrect, and valgrind giving no leaked memory. However, because I was working to the last minute, when I made some final changes to throw `EXIT_FAILURE` on errors, I deleted a necessary line to save my beginning node. I had to submit the assignment before the deadline had past, and I couldn't realize my mistake that lead me to miss the freeing of that last linked list of data and leaking memory.

## Assessment

My assessment plan was to step by step test and check all my functions as I went along. Using print statements to stderr and color codes, I made sure that each function outputted exactly what I wanted and checked often to make sure. Only when all the necessary functions worked correctly did I use the given inputs to run with and check against the given outputs. I used `testdiff` to check for differences between the files and `gdb`, `valgrind`, and selective function checking to ensure that all my code worked along the way, only implementing bite sized pieces at a time. By the end, however, I realized a major bottleneck in the time I took to test cases. With two different versions of five different input files with six different outputs each, there were a lot of files to check for and check against, and it quickly became tedious checking each of these. I had to adapt my Makefile to use placeholder variable to reduce the amount of duplication and save time, but even then it was still a tedious task.

## Next Time

Two major goals I still have to work on for PA3, and one ongoing goal. The ongoing goal which I always need to try for is starting early and doing little bits at a time. I tend to work in large chunks for longer amounts of time, and then things might end up close to the deadline. I plan on resisting this tendency and spreading out my time to reduce stress on myself and to increase the quality of my work. On that point, another thing I will be sure to do for next assignment is to ensure I have ample time before I submit. I want to make sure that the small issues I check before submitting, last time functionality checking, `valgrind` checking, all files properly zipped up, is able to run on the makefile compile command or the default one, being able to work on my own machine. These are all things that I should check with ample time. I also want to plan my tests and their makefile commands better, so that I don't have to waste as much time writing them.