# Contributions

Everything in the assignment was done by me with some references to stackoverflow answers, the reference links are in the source code.

# Problems Encountered

For the first version of the programs I used linked lists for both programs and the logic was easier to implement and it only requires a single read of the files. However, the speed was a huge issue and linked lists can’t really be multithreaded, so I changed the generator program to use arrays instead so that I can multithread the program. This requires the program to read through the file twice, once to get the length of file, so I can allocate memory for the arrays and 2nd time to store data into the arrays. Reading the file twice make negligible difference as it is fast and not the most computationally heavy part of the program. After implementing multithreading, the runtime decreased from 6 minutes plus to 48 seconds.

For the first version of the recovery tool which used linked lists the runtime was about 20 minutes. After I change it to use arrays and improved the algorithm, the time decreased to about a second. This really blew my mind as the improve in algorithm in theory should have only increase the speed by 50%. I suspect it has to do with the memory allocation of arrays and malloc.

# Assumptions/Limitations/Bugs/Issues

Assumption is made that the user will use lookup table generated by the generator I wrote. So, the recovery tool will only work as intended if the lookup table is generated by the generator or if the lookup table is in the same format as the output generated by the generator.

Both programs rely on the file command to check for file format, if the user does not have the file command, the program will fall back to a simple extension check which can be easily bypassed. Through my testing the generator program seems to be safe, however the recovery program will have segmentation fault when the extension check is bypassed, and a binary file is passed in. The user will be recommended to install the file command. However, this situation is unlikely since file command is pretty much present on all distributions.

For the generator program there is no option to choose the number of cores to use and the program will default to the maximum number of cores. This can be an issue if the user is multitasking as it can cause lag.