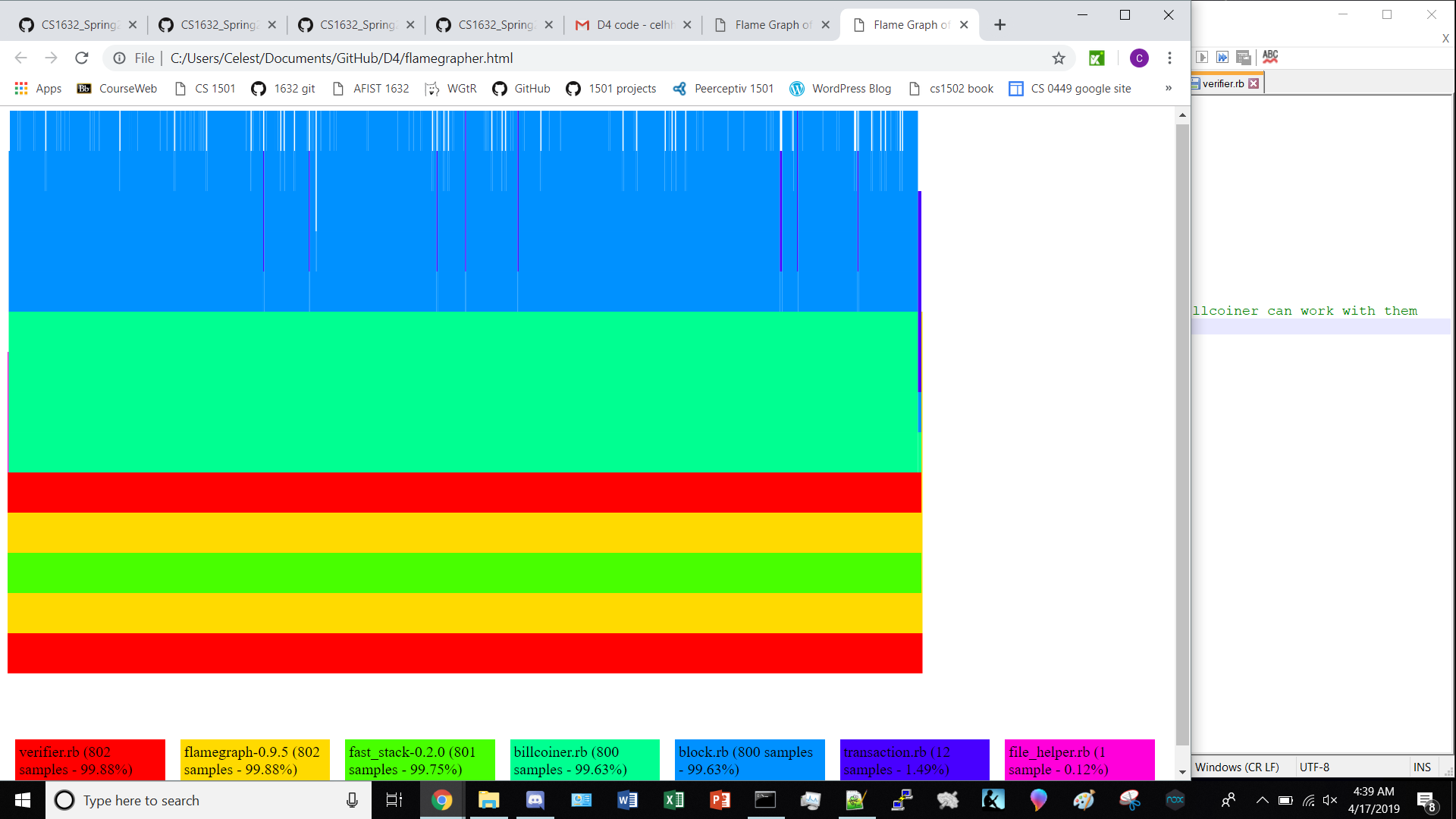
CS 1632 – Deliverable 4: Performance Testing

|  |  |
| --- | --- |
|  | Celest Hayden  chh147 / celhhayden |
|  | https://github.com/celhhayden/D4 |

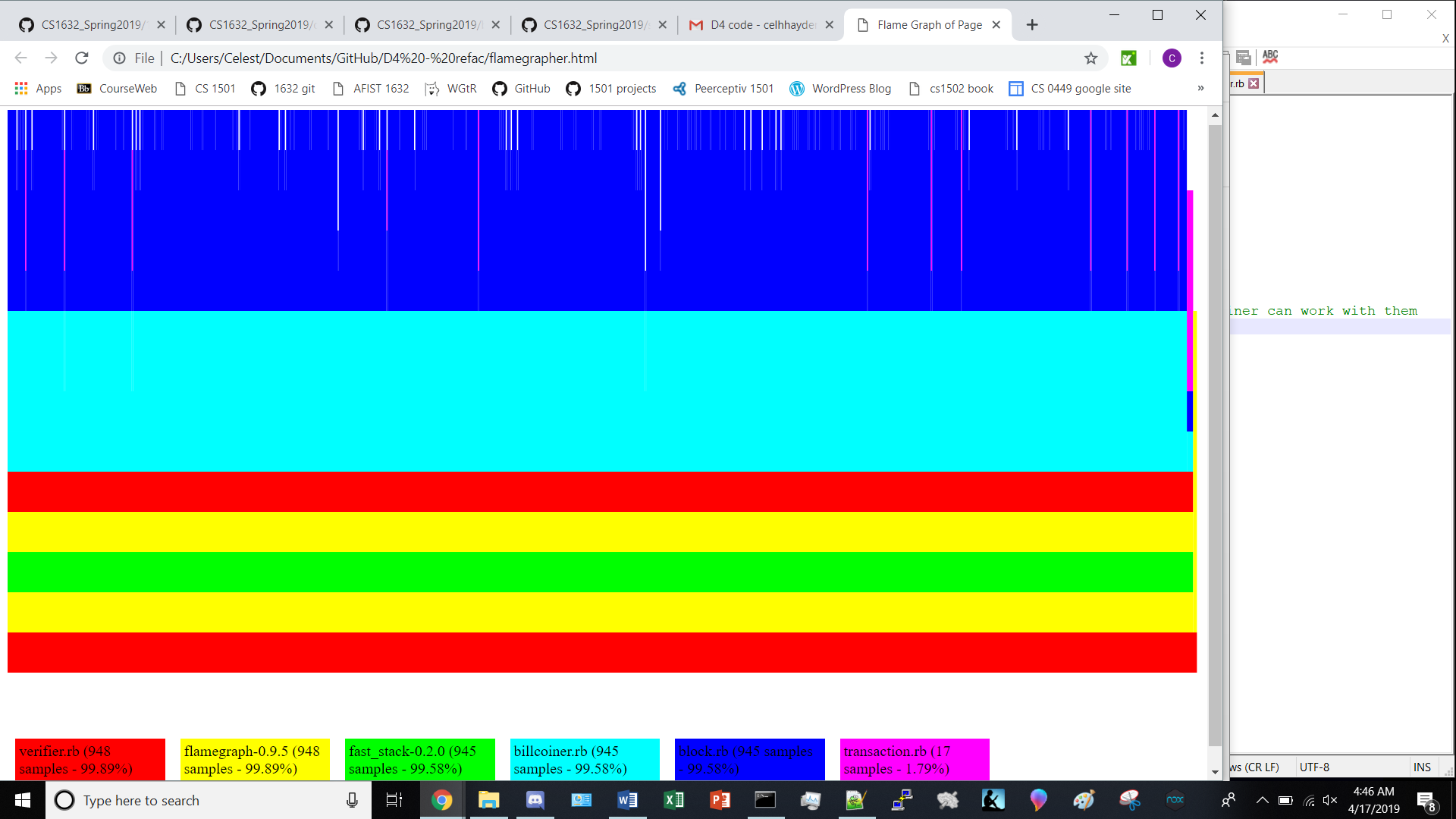
This was certainly an interesting project. The way we approached it was by splitting up the initial code writing. Specifically, building the block and transaction classes that handled the actual input, and the verifier, file reading. This caused a bit of issue since we both had our own style of coding. We also had slightly different takes on how to handle the classes, both inside and out of them, so even before putting everything together we had to do a bit of refactoring. But, once we both got to see each other’s work we managed to pull everything together and iron out any kinks we couldn’t test for yet. After getting the code working we tested long.txt, but the flamegraph wouldn’t load afterwards so we went with 1000.txt instead for the graphs. After that, the real hard task was getting the refactored program down to a decent time. The biggest issue was around the hashing function, as we saw on the flamegraph. Sadly, we had no idea how to reduce the time it took to work with large hash values, so there is little to no improvements in time. The only changes made in an attempt to reduce the time it took were reducing certain lines of code. Also, using a hash table structure instead of an array for certain objects may have also affected runtime. Arrays are slightly easier to access, so using them would improve runtime for large files, like long.txt. We were not able to focus much on unit testing, so there are only a few for block.txt.

**Based on 1000.txt**

**BEFORE**



**AFTER**



**Times for long.txt**

**BEFORE**

|  |  |
| --- | --- |
| REAL: | MEAN: |
| 1m21.526s | 1m11.657s |
| 1m4.593s | MEDIAN: |
| 1m9.232s | 1m9.232s |

**AFTER**

|  |  |
| --- | --- |
| REAL: | MEAN: |
| 1m15.301s | 1m10.612s |
| 1m6.562s | MEDIAN: |
| 1m9.973s | 1m9.973s |