Analysis of CPU scheduling algorithms

\*In the attached files you will find a spreadsheet containing all of the data produced by source code (also included)

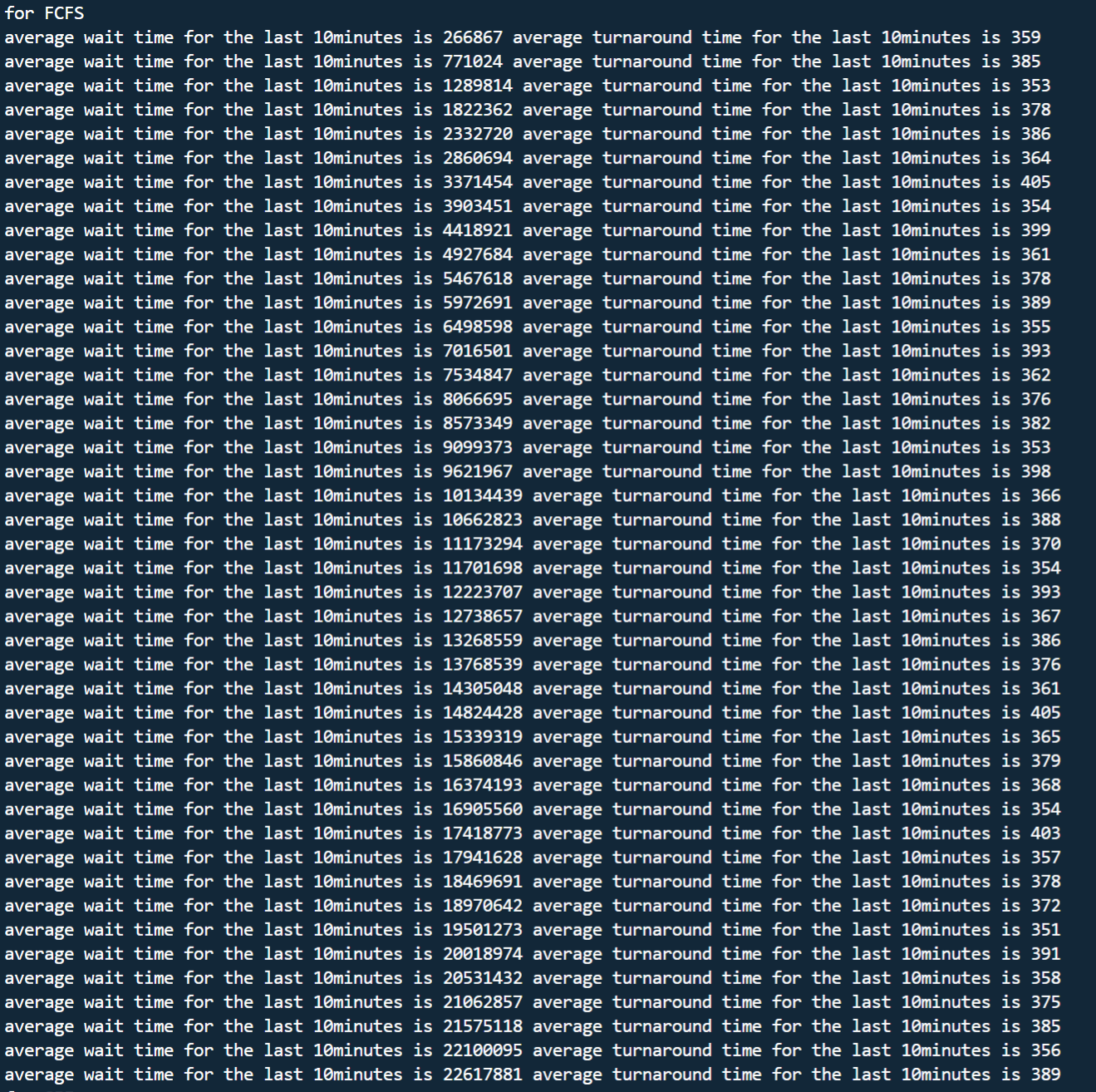
It can be seen in figure one that the first come first serve algorithm had the fastest average turn around. The wait time caused by FCFS is difficult to see in Figure 2 since it is almost exactly under round robin with a time quantum on 8. Since FCFS does not follow any prioritization of the processes the average wait time stays relatively consistent and does not suffer from a spike in wait time caused by starved processes. However, comes at the price of causing short processes to wait for longer processes to complete.

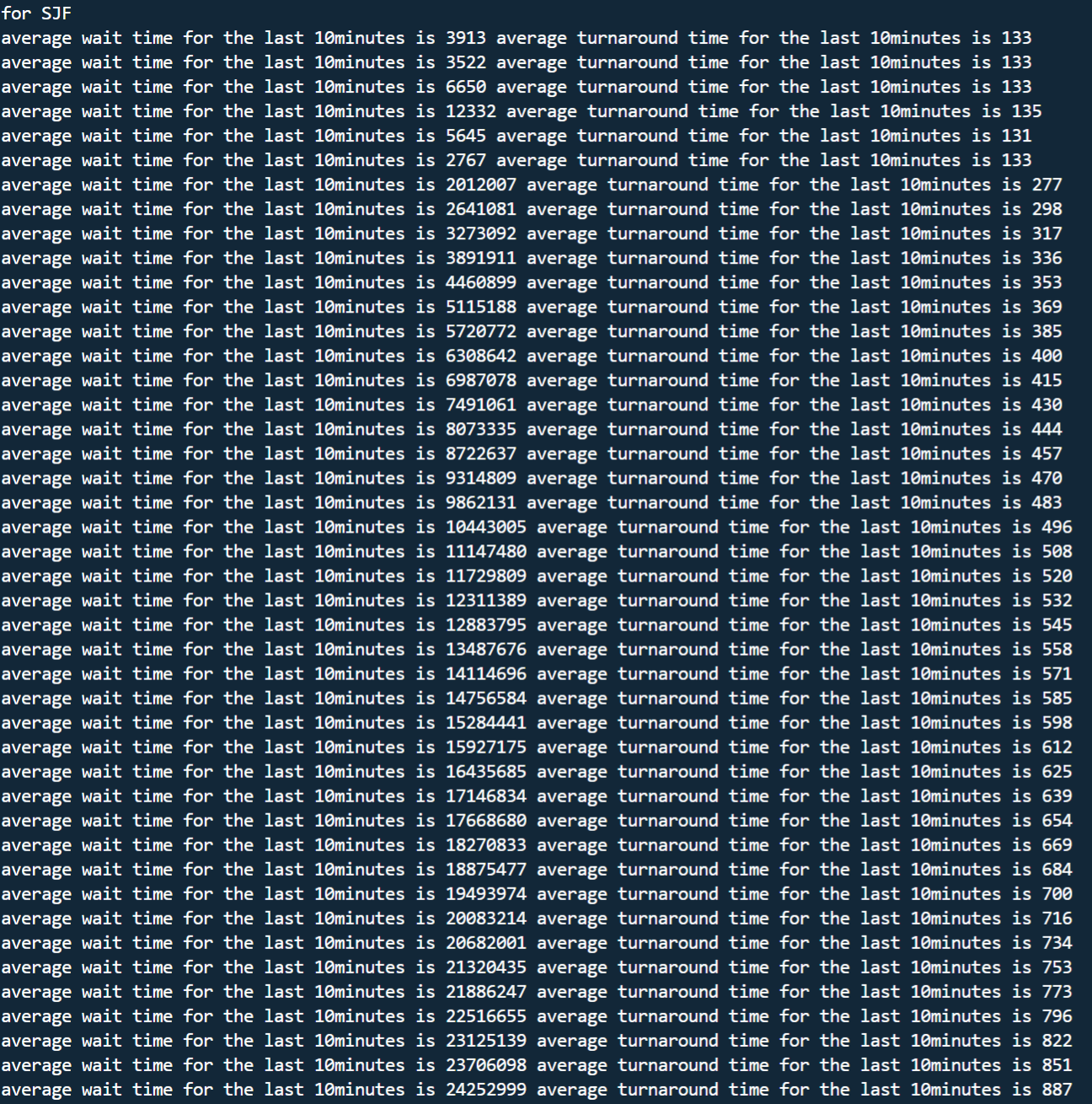
To the slow throughput in FCFS, the Shortest job first algorithm can be used. SJF had the fastest initial turnaround with the second lowest average turnaround. it is nearly not even visible in the graph. This was expected since the largest advantage of this algorithm is its fast throughput at the cost of possible starvation of the larger processes causing the large final wait time seen in figure 2.

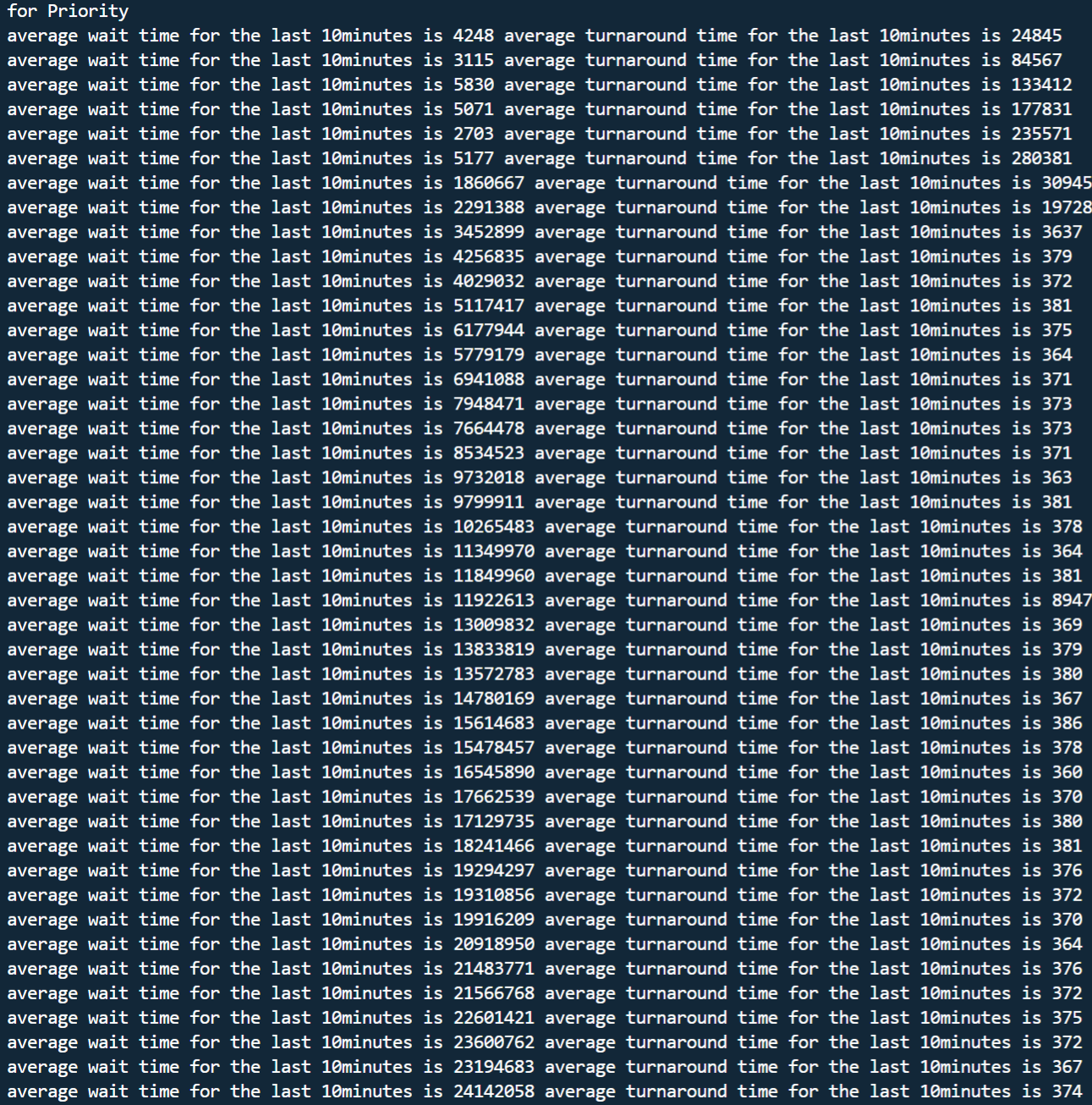
When comparing SJF to Priority they almost act in opposing turnaround times. Initially, the average turnaround time in Priority is much slower since the processes are interrupting each other when a higher priority process arrives. However, the turnaround time eventually falls once all of the higher priority processes are complete. Something that both priority and SJF have in common is their long wait times at the end of the test seen in figure 2 caused by the starved processes with lower priority.

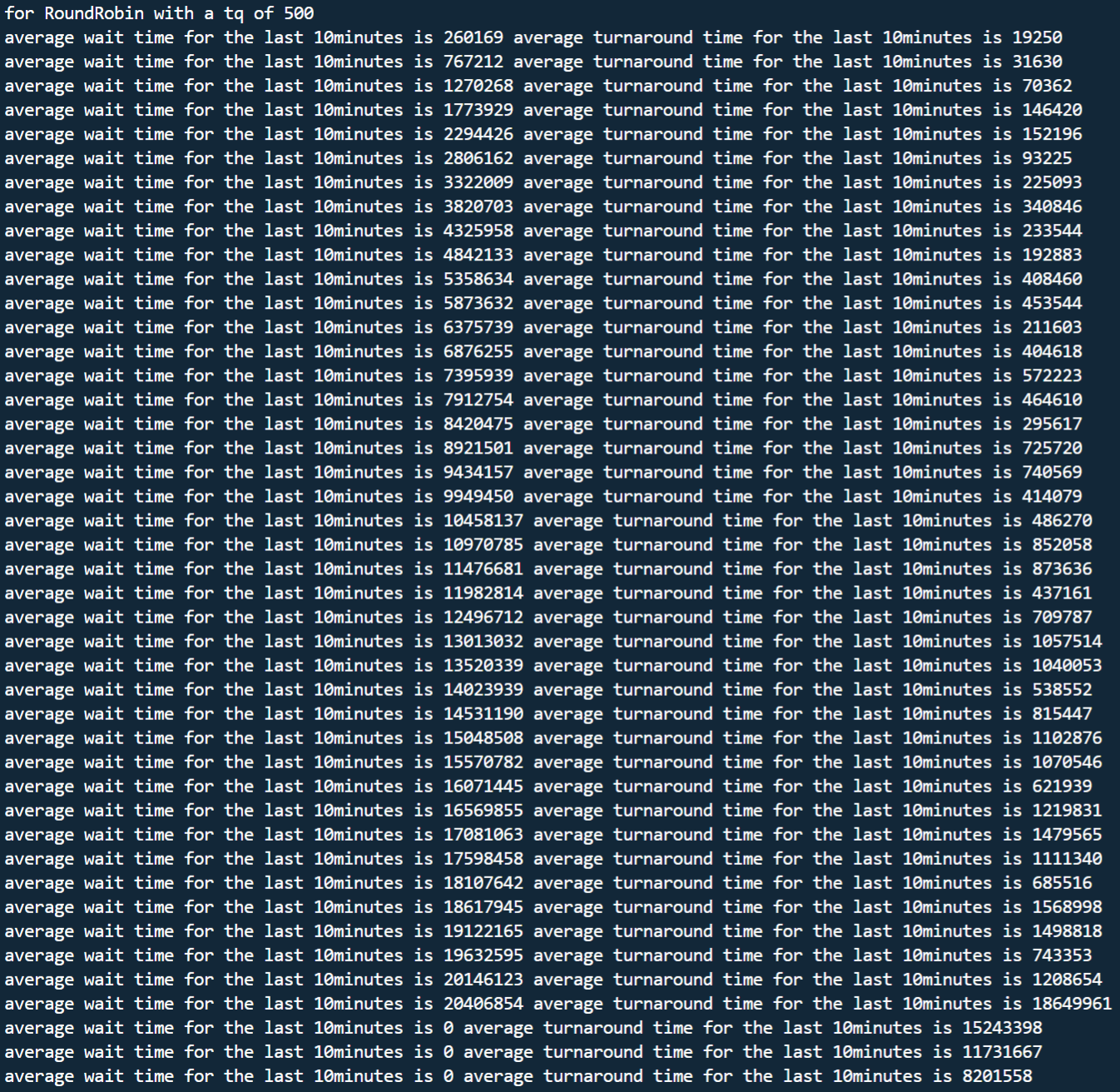
Finally, we see the round robin algorithm which was tested with 2 different time quantum values (500 and 800). For the time quantum of 500, the turnaround time becomes horrendous since the processes are constantly being swapped with one and other due to the low time quantum. For the TQ of 800, the turnaround time is much better due to the higher TQ allowing the smaller processes to complete before the TQ swaps the process. However, both options demonstrate inconsistent turnarounds and tend to rise in turnaround time. When comparing wait times however the TQ of 500 allowed a much lower wait time when compared to the TQ of 800 which was extremely similar to the wait time of FCFS this is due to the average CPU burst being 1000ms making the TQ so close to the average causes it to act similar to FCFS when a processes burst time is below the average.

## Screenshots of running code

This is the working coutput for the First Come First Serve algorithm.

This is the working coutput for the Shortest Job First algorithm.

This is the working coutput for the priority algorithm with a max priority of 30.

This is the working coutput for the Round Robin algorithm with a TQ of 500.

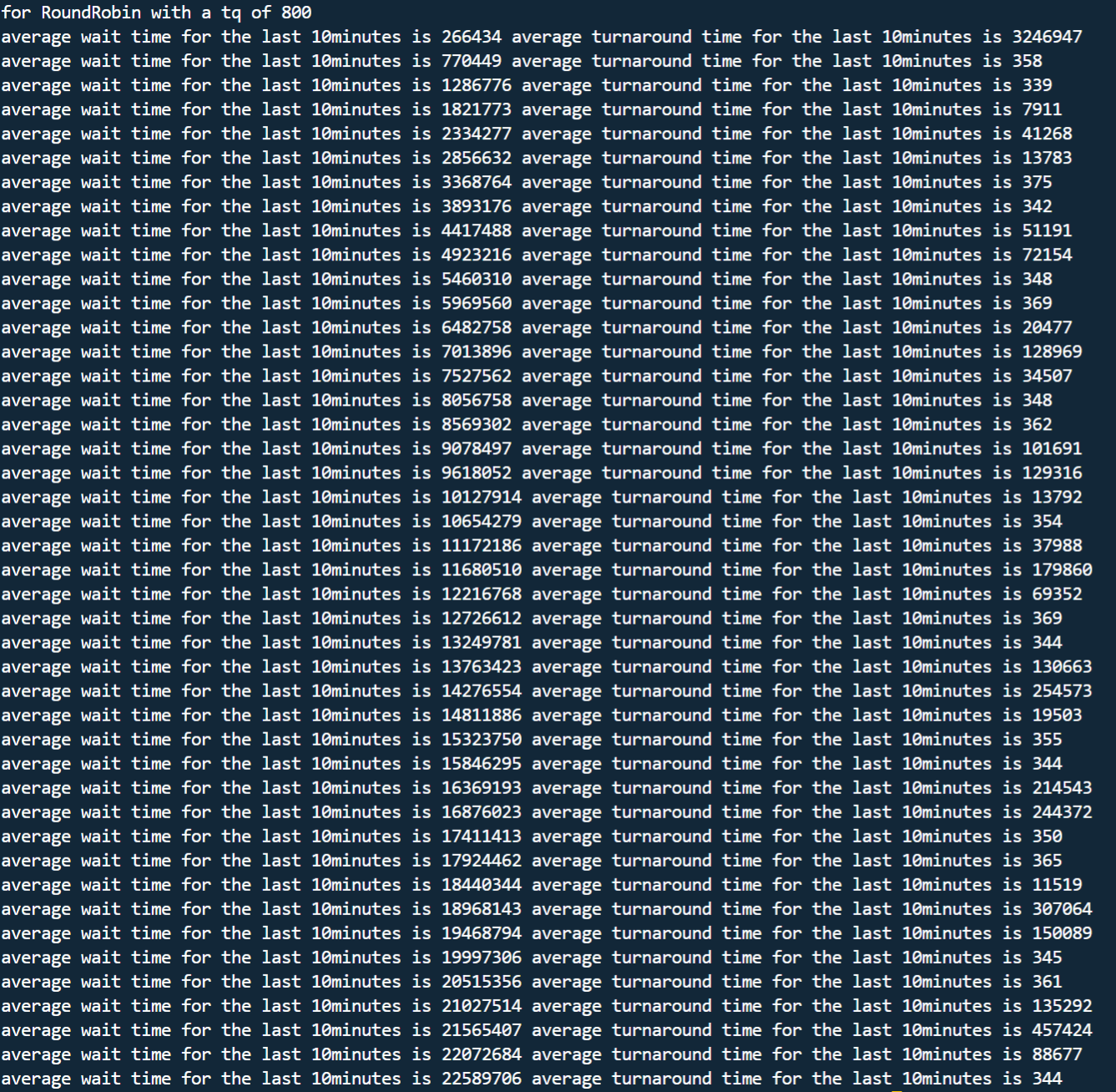
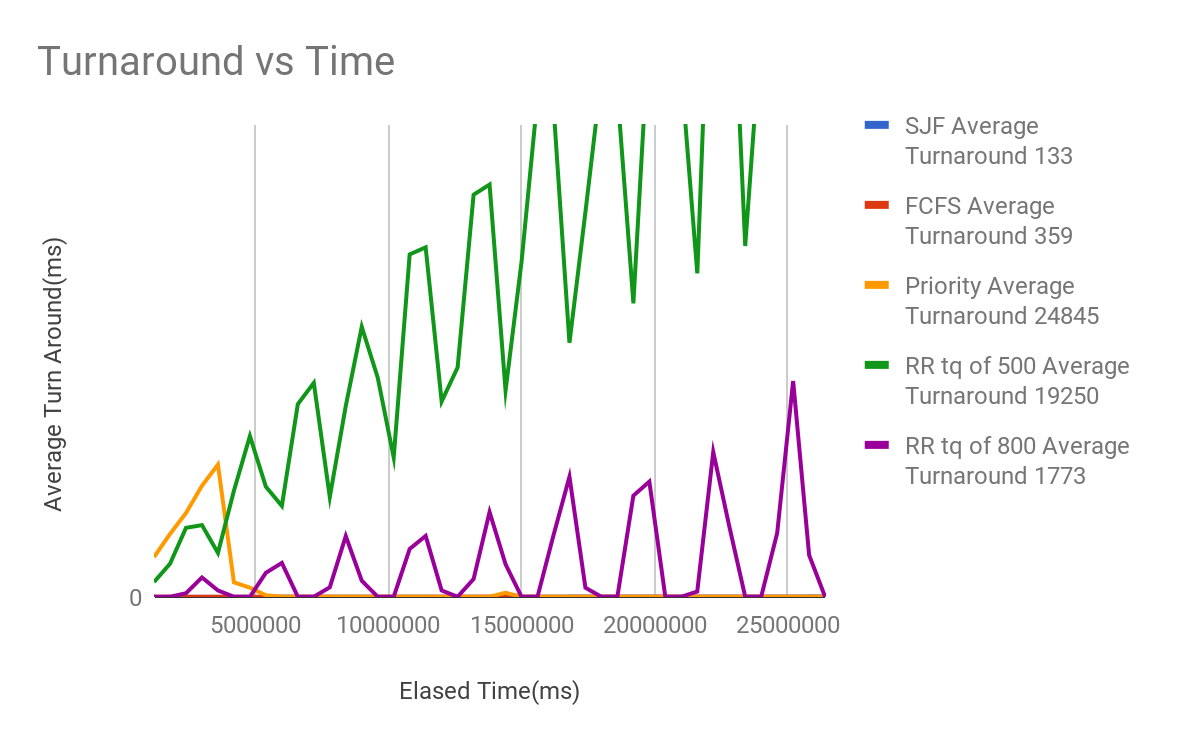
This is the working coutput for the Round Robin algorithm with a TQ of 800.

Figure 1

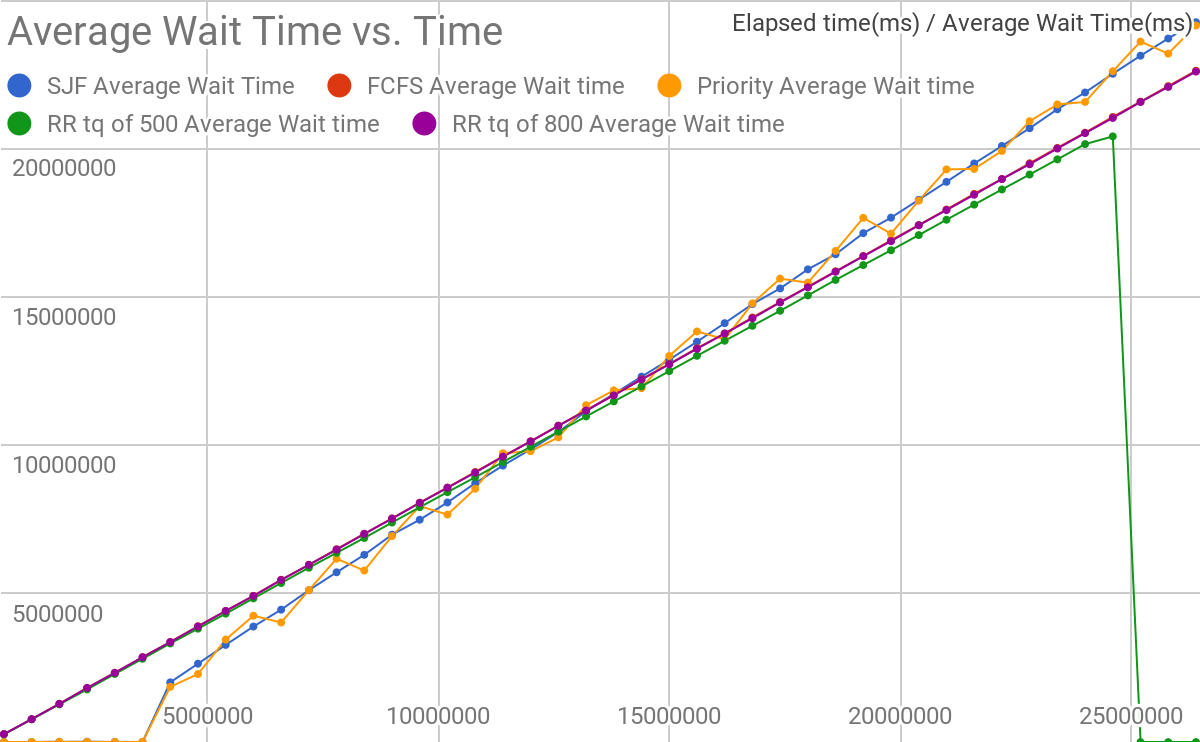


Figure 2