Term Project Observations

Timothy Flucker

MET 526 – Data Structures and Algorithms

For this project I used an ArrayList data structure to contain the data that was read from the input file. This data was then filtered and the object with the minimum priority was chosen and removed from the list and added a Java PriorityQueue. A custom comparator was created so that my user-defined “Process” objects could be differentiated and sorted by their “priority” field value.

One observation that I had for this project was the usefulness of the comparator for the custom “Process” object I created. It was so useful for quickly picking the minimum value in the Java PriorityQueue object, and I had used it previously. This functionality allowed me to simplify my code and rely on the characteristics and ADT of the PriorityQueue to consistently remove the minimum value based on the definition in my Comparator.

Another observation, was how long certain process object waited in the PriorityQueue. In the input file each process only has values between 9 and 25 for their durations, however while the code is running some processes waited as long as 80 or 110 iterations. For the proposed implementation based on the pseudo-code, this output makes some sense, however it also highlights how this implementation could be improved. One way to improve it would be to multi-threading and concurrency so that processes can run in parallel. While this might be more expensive in memory usage, it would allow processes to be completed in a timelier manner.

Another observation was how playing with the ‘maximumWaitTime’ value can change the results. Using the default 30, it takes the system 390 iterations to fully process the data. However, if that value is adjusted to be less than 15 or greater than 35, then it takes the system 391 iterations. Additionally, if the maximumWaitTime is modified to be 93 or greater, then it takes the system 402 iterations to complete. It is interesting to note how adjusting this value can cause different outputs.

In this project I learned about how to use a Java PriorityQueue as well as how to define and utilize my own custom comparator. Additionally, I got to practice with Java Streams to efficiently filter my ArrayList and compute the average wait time value. Everything else related to this project was familiar based on my previous experience working with Java, as well as my other courses in my graduate program.