

Junnan Shimizu

Project 9 - Disk Scheduling Algorithms

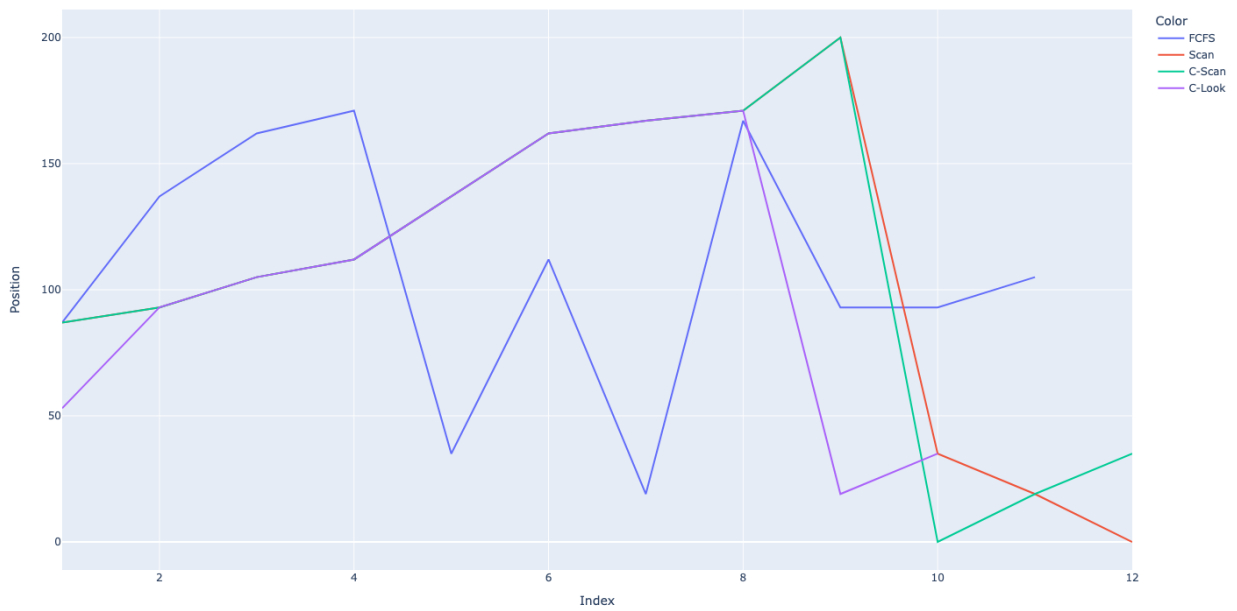
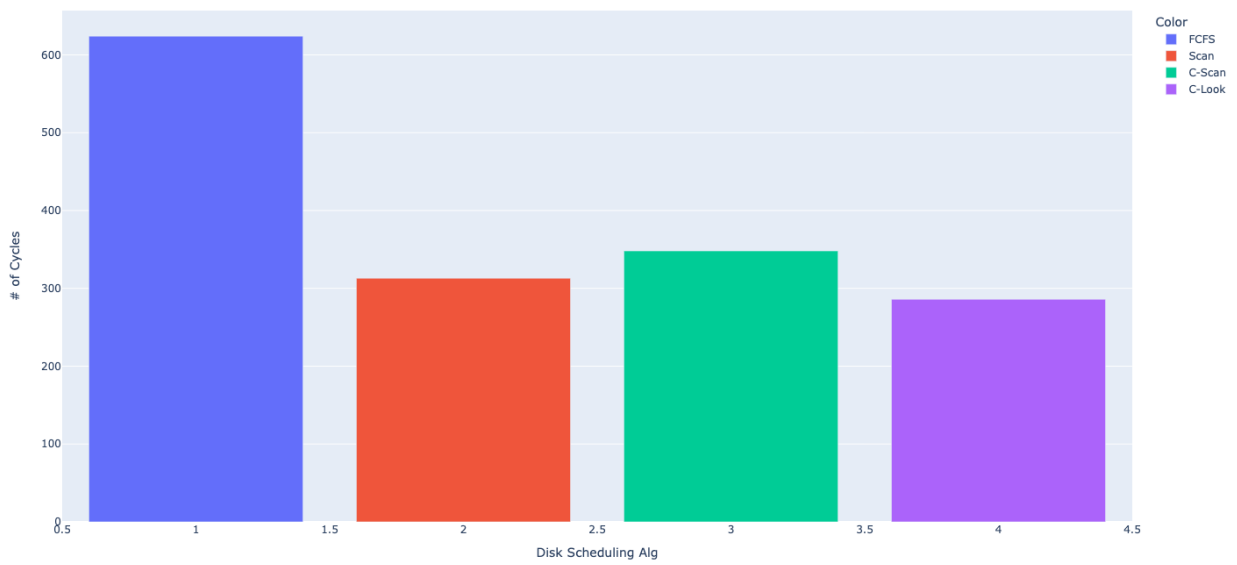
Abstract:

In this project, my goal was to visualize how disk scheduling algorithms worked using python's visualization library. In this project, I tested first come first serve (FCFS), scan, C-scan, and c-look disk scheduling algorithms. We compared each of the scheduling algorithms against each other using a simulation.

Results:

Example Run 1:

```
Start: 87
Data: [137, 162, 171, 35, 112, 19, 167, 93, 93, 105]
FCFS Cycles 624
Scan Cycles 313
C Scan Cycles 348
C Look Cycles 286
```



Example Run 2:

Start: 45

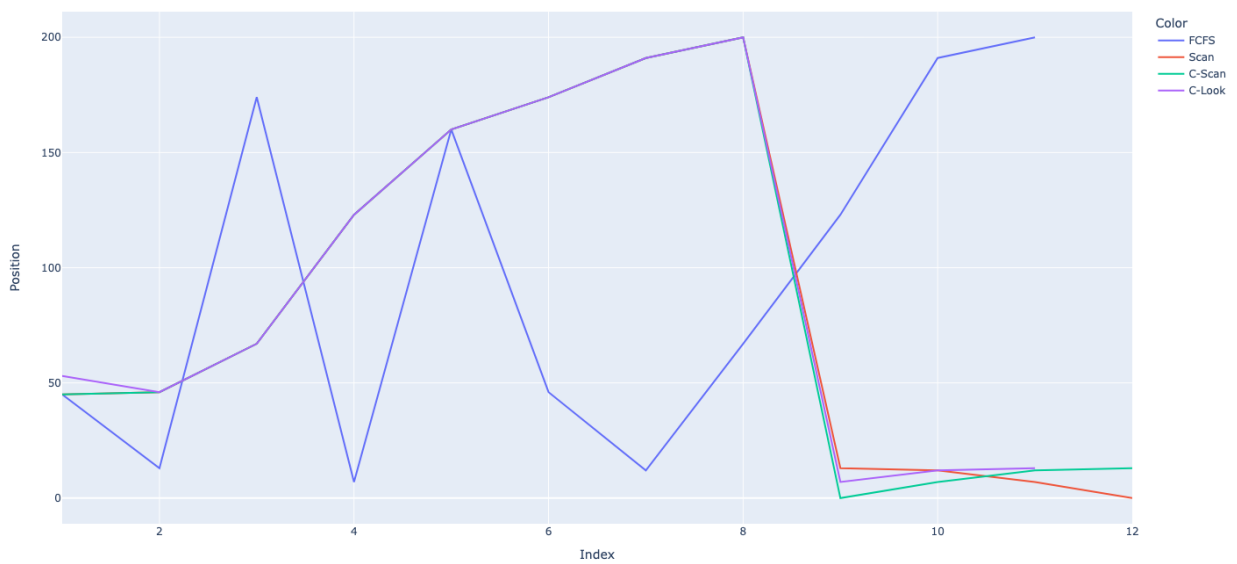
Data: [13, 174, 7, 160, 46, 12, 67, 123, 191, 200]

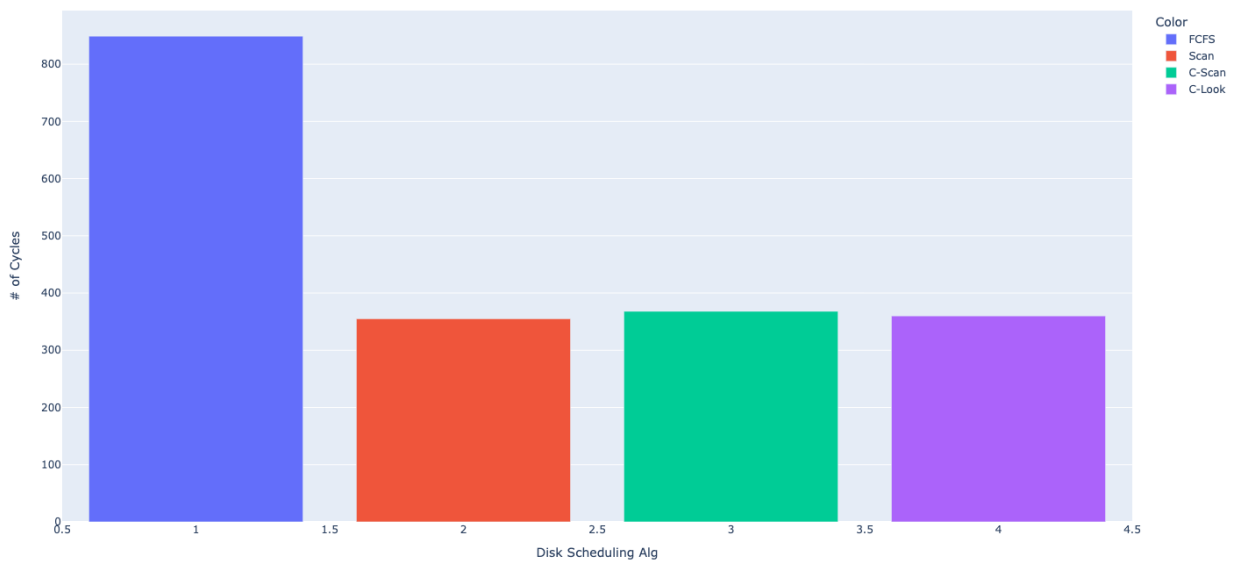
FCFS Cycles 849

Scan Cycles 355

C Scan Cycles 368

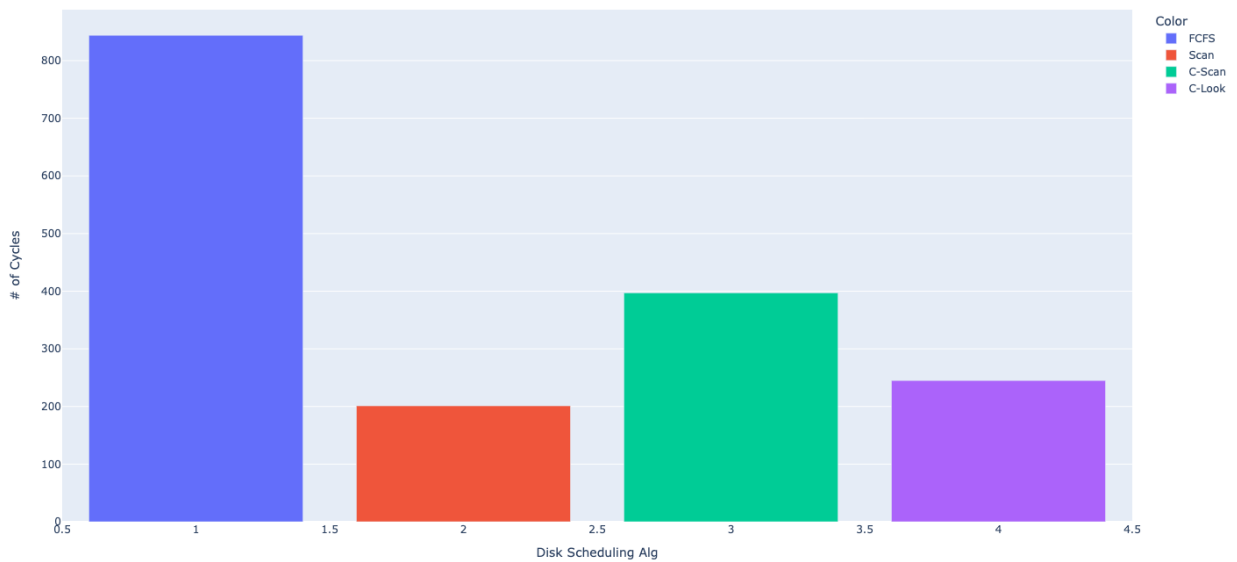
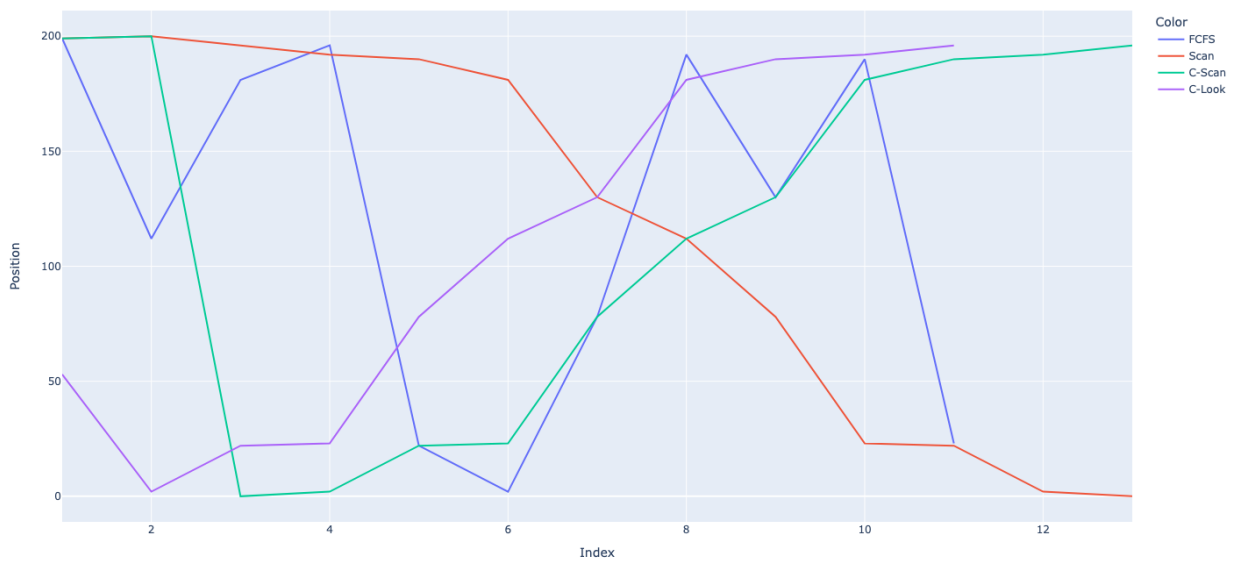
C Look Cycles 360





Example Run 3:

```
Start: 199
Data: [112, 181, 196, 22, 2, 78, 192, 130, 190, 23]
FCFS Cycles 844
Scan Cycles 201
C Scan Cycles 397
C Look Cycles 245
```



After running the test about a dozen times, it is clear that first come first serve is the least efficient of the four algorithms (however, there was a case where it beat out the others). The other three algorithms vary a ton, and not one consistently beats out all of these others with the random data sets we are utilizing.

Discussion:

All these disk scheduling algorithms work best in certain situations, and even first come first serve, the simplest of the bunch, outperformed others in very unique situations; however, we learned that shortest seek time first, usually outperforms all of these consistently.

Extensions:

I did not do any extensions for this project.

Acknowledgements:

I did not get help from anyone for this project. I did however reference this website for help with python visualization: <https://www.geeksforgeeks.org/data-visualization-with-python/>