

Of the concepts we discussed, which one(s) did you find most interesting or useful, and why?

For me, the most interesting task to explore here was exploration on PageRank. Even if the concept itself is applicable in much narrower field than some of the other topics, the foray into multiprocessing, MapReduce, and refreshing on linear algebra mechanics has made the project extremely interesting.

As a corollary, I was able to explore Eigen, the linear algebra library. This allowed an additional degree of understanding that grounds the design pattern of other tools such as Numpy, which is based the design of Eigen.

What concepts did you find most challenging, and why? What did you do to overcome these challenges?

The most challenging concept for me was actually Finite State Machines proofs. I wanted see how the mechanisms for FSMs applied in other areas, which is actually surprisingly difficult to identify: components like complex invariants, or proving how they evolve in some cases. However, I believe that the skills of proofing will be very useful in developing further mathematical and CS skills in the future.

Were there any concepts that you found to be particularly easy to understand? If so, how did you challenge yourself?

I think the Dynamic Programming section was particularly easy to grok for me; perhaps because some of my previous experience writing competition programming programs, for which DP often plays a great part, I had a regimented idea of what DP is and can do.

However, what made string distancing a nontraditional problem is that its a multiple-action graph based DP problem, and I also endeavored to refresh my memory of high-performance programming by writing it in the language where it would usually be written: C99.

What are some ways that you displayed good habits of mind or contributed to a good learning environment in the class?

I consistently tried to help those in the class, supporting my table group in problems and working through challenges. Furthermore, I consistently tried to challenge myself by using unknown tools or methods I have not used frequently to refresh my knowledge of industry methods atop of the content areas.

What was the most useful or memorable piece of feedback you received this semester, and how did you act on it?

Despite the repeated challenges, I loved the process of revising my PageRank assignment, and—in process—discovering the quirks of C++. I am very thankful for the feedback I received in revising the assignment and to discover—as if collaboratively pair programming—mistakes and improvements I can make next time I am implementing an algorithm or was writing more C++.