Jacob Christiansen
Career Project
CCCO Summer 2018
MAT 202: MA1

Computer Science and its Relation to Mathematics
and
Interview with Aaron Morneau

A description of Computer Science and related fields:

Computer Science is a very broad subject, but can be broken down into a few specific categories: IT (information technology), Research, and Development.

In IT, employees provide support on specific products or general technology used by a company. Usually they are called upon when questions arise about said technology, or sometimes they just need to provide scheduled maintenance. The day-to-day tasks of an Informational Technician vary more than any other job in the Computer Science career pool (Modis, 2018).

Doing research in a Computer Science field means that it's your job to create, test and analyze different programs and networks for informational purposes, and to further the advancement of technology. Finding solutions through this type of work can change how the rest of the world works and lives. Because of Computer Science researchers, we see such radical advancements in technology so quickly (Modis, 2018).

Development in a Computer Science field usually means that you will be creating and working on software products for a company. Most companies need specific programs to fit their personal needs, or the needs of consumers (in which case they sell software), and require Software Developers to create them. Hardware Development also is covered in Computer Science, however that tends to lean towards Electrical Engineering careers (Modis, 2018).

Summary of a phone interview with Aaron Morneau, employee of UNH's InterOperability Lab:

Aaron Morneau is someone I met during my high school years at a "Job Show", where my school brought in people to talk about their careers. I ended up getting to know him better by shadowing him in 2016.

I began the interview by asking him what he does on a typical day of work. His response was, *"To simplify it, I generally do IT work, and little bit of coding on the side"* (Morneau, 2018). He then explained how most of the time he was going back and forth between his desktop and the IOL's server room.

I proceeded to ask what kind of work he does with the servers. He explained that most of the IOL's data was stored there, and that he has to ensure it's integrity.

"I have to just make sure we don't lose all that data by keeping the servers happy" (Morneau, 2018).

"I didn't know servers could have emotions," I responded.

"Trust me, they do, and you can tell when they are angry" (Morneau, 2018).

I then asked what his favorite part of his job was. Aaron stated that he really enjoyed working with a group. Apparently in IT, you don't get to work with other as often as he likes, or at least in his specific case. Next, I asked him what his hopes were for the future in his career. *"Not that I don't like it here at UNH--I really do--, I just have this urge to move on. I mean, I've been here for all of my undergrad, and now almost all of my masters, but once I graduate I hope to move to a larger company that allows for larger projects, and hopefully better pay"* (Morneau, 2018). I can relate to Aaron in this case, I definitely know the feeling like you've been in one place for too long.

Finally, I ended the interview with the question, "What advice would you give to a student studying your field?" His response was very helpful for me, I believe. Aaron said not to get scared if you no idea what you are doing. "*The truth is, not many people in this industry do. Schooling can only prepare you for so much"* (Morneau, 2018). He also advised to get as much experience as possible, as that is what the industry will look for when hiring.

A Computer Science related math problem:

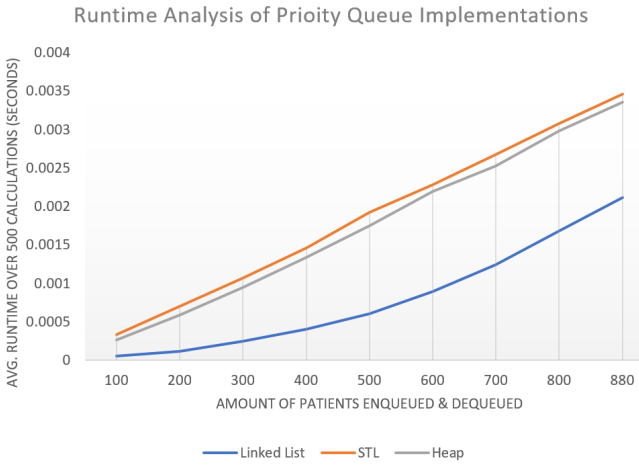
In my CompSci II course in Spring 2018, I had to program a way to enqueue/dequeue 500 "Patients" (our name for a node) using three different methods, STL, Linked Lists, and Heaps. A write up of the project is accessible via the Works Cited for reference (Christiansen, 2018).

This is the data I collected:

Amount of Patients Enqueued/Dequeued

Implementation	100	200	300	400	500	600	700	800	880
Linked List	0.000054 seconds	0.000120 seconds	0.000246 seconds	0.000408 seconds	0.000610 seconds	0.000890 seconds	0.001246 seconds	0.001676 seconds	0.002116 seconds
STL	0.000334 seconds	0.000706 seconds	0.001068 seconds	0.001458 seconds	0.001924 seconds	0.002278 seconds	0.002672 seconds	0.003076 seconds	0.003456 seconds
Heap	0.000266 seconds	0.000590 seconds	0.000948 seconds	0.001338 seconds	0.001748 seconds	0.002196 seconds	0.002526 seconds	0.002978 seconds	0.003354 seconds

From left to right, is the average runtime for 500 tests on x number of *Patients* enqueued and dequeued, while from top to bottom is the different implementations used.



As you can see, two methods show linearity, while the one is exponential. In this case, the smaller runtime, the better, therefore Linked List is the fastest. But that will not be the case forever, as LL eventually will become slower than the others as the number of data nodes processed goes up. I was curious to see at what point that would be, so I turned the data into estimated sequences!

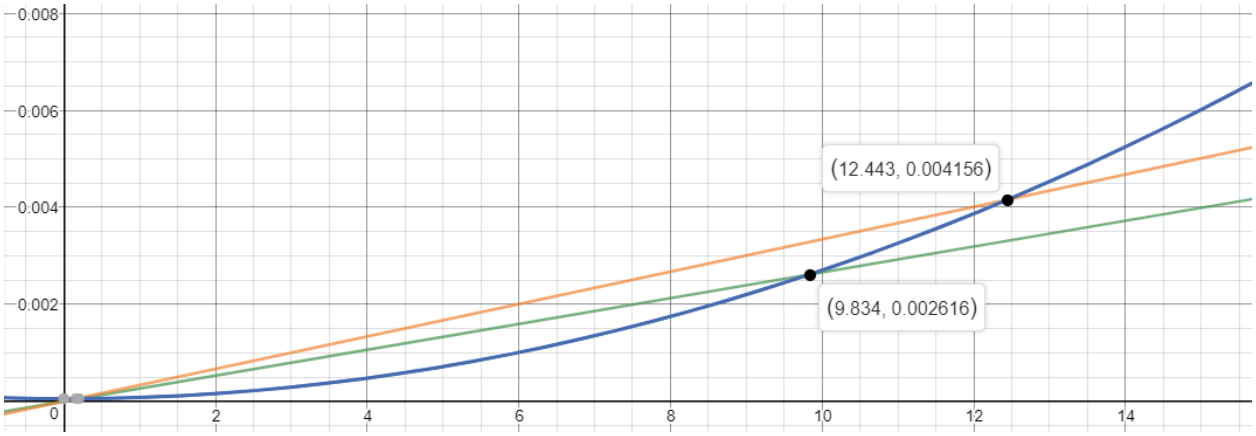
Linked List: $a_n = 0.000054 + (n * (n/2))(0.000054)$

STL: $a_n = 0.000334 * n$

Heap: $a_n = 0.000266 * n$

I did this work by using the data I already had as terms in the sequences. Unlike problems we've been given in this course, this sequence doesn't have an exact answer, and the points were

based on many changing variables, similar to how most things are not exact numbers in “real life”. STL and Heap were easy because they clearly were very close to being linear, by just doubling each time. However Linked List was a bit trickier, it wasn’t just $0.000054 * n^2$, that was too fast of an increase. So I tried many things to acquire the proper increasing rate like $n - 1$, $n * (n - 1)$, and eventually found one that matched my data points pretty well, $n * (n/2)$. So now that I have my new equations, I can graph them:



Note. n=1 is equivalent to 100 patients on my previous data.

Now we know that Linked Lists becomes the slowest method at approximately 1244 data nodes.

Works Cited

Christiansen, Jacob. An Analysis of Priority Queues with C++. University of Colorado at Boulder, May 2018,
www.scribd.com/document/384961099/An-analysis-of-Priority-Queues-with-C-by-Jacob-Christiansen?secret_password=Cei21fWsMurJ59xgnl9J

Inc., Modis. “Detailed IT Job Descriptions Across the Industry.” Modis, Modis, Inc., 2018,
www.modis.com/clients/salary-guide/job-categories/.

Morneau, Aaron. “Computer Science as a Career.” 15 July 2018.
aaronmorneau94@gmail.com