Summary - Shor's Algorithm

Elliott Pryor

2 December, 2019

Our project was done on Shor's Algorithm. I personally really enjoyed the project it was a lot of fun, and exciting to learn about how to leverage properties quantum computers. I think the quantum algorithm was very interesting and important due to recent developments of quantum computers. Overall, I believe that we did a very good job working together as a group (except the minor mishap with P2). We had a lot of communication through text and used git to sync our work so we were all on the same page.

It was very challenging to do much with Shor's algorithm besides describe how it works because we can't run it on a quantum computer. Therefore we just had to explain the logic and work through some examples. This helped me get good idea of how it works at a very high level. We also wanted to prove some of the theorems that we use for period finding. For example: it would be very cool to prove why $f(x) = g^x \% N$, where g and N are co-prime, is periodic. I attempted to prove this at one point; however, I really had no idea how to approach this problem and couldn't figure out how to do it. I think that the classical implementation was good and it shows what happens in a computer, but it wasn't amazingly difficult. I had assumed that it would be much harder to implement classically. It also doesn't provide a lot of insight into how the quantum algorithm is superior to the classical implementation.

I think we did a very good job of conveying how Shor's algorithm works. The actual components of the algorithm are very simple: guess a number, make sure it isn't a factor, find the period, then use Euler's GCD formula for the factors. When researching this, these very basic components were hidden behind a lot of math and quantum state notation. This makes it very hard to parse and understand what is happening. We provide an easy way to understand the basics of the algorithm without over-complicating it or using foreign syntax.

This also helps bring quantum algorithms to the attention of our peers. While speaking with peers about our projects, I mentioned that I was working on a quantum algorithm and they assumed it must be incredibly complicated and beyond their abilities. My peers knew from the media that quantum computers could supposedly break encryption and factor numbers very quickly, but did not know the algorithm behind it. The purpose of this project is to expose our peers to some of these technologies. As quantum computing becomes more prevalent it is important to be educated on these issues. Our project does a good job of exposing peers to these issues in a very digestable manner so that they can be more educated and limit the spread of misinformation.

I did lots of the base work in finding good sources for the algorithm and typing up how it worked in the earlier parts. So almost all of the description of how the algorithm works in parts 1-5 was done by me. This gave us the foundational knowledge that we needed moving on. This resulted in reading many papers and watching several videos on how Shor's algorithm works until I felt comfortable that I could easily explain it to anyone. Honestly, this was my favourite part as it was very interesting and I learned a lot about Shor's algorithm and other related subjects.

It was also my job to work on the classical implementation of Shor's algorithm. I originally thought that this would be much harder than it was. I used python so that we could handle very large integers without overflowing. Eucid's GCD algorithm was taught in my CSCI246 class so I could use my notes from that class to easily type the function for that. I just did a brute force period finding algorithm that just counts until it finds a repeat. I do not know of any faster way to do this classically. The only issues I had were in recursion so I had to change euclids algorithm to be non-recursive and I made some improvements to the period finding algorithm so that we didn't calculate the exponent from scratch every time we just multiplied the previous value by the base of the exponent (ie 3^i we would multiply by 3).