CIS*4010 A4 — Jason Nguyen

My project takes in a graph as input and attempts to generate a solution to the graph colouring problem using that graph configuration. It then exports it in Graphviz DOT format so that users can actually get a visual representation of the solution. I adapted this code from the dwave-examples repo, cleaned it up a bit, added more debugging statements and finally I added an import/export system. This augment allows the user to represent a graph as a series of edges in a text file and then run the program using these input files (the original code used an unintuitive, hardcoded example). Using the pygraphviz module, I made the program export full-colour diagrams instead of the original bland text-based output (e.g. "{1: 0, 2: 1, 3: 1, 4: 2, 7: 0, 8: 2, 5: 0, 9: 2, 6: 1, 10: 0}"). An example can be seen in the README.

I chose this project because graphs have interested me since coming to Guelph. At first I thought they were stupid flowchart-like diagrams that computer scientists turned into jargon for the sake of jargon, but as time went on I fell in love with them. Particularly in CIS*2910 when Prof. Sawada would tell us about all of the different ways you could view a graph, model real-life problems with one, and the like. Graphs have worked their way into my personal projects too. This summer, I worked on a UoGuelph dependency graph generator that takes in all of the school's courses and creates DOT graphs modelling the courses' prerequisites (example). Naturally, when I saw DWave being mentioned as a potential A4 project and noticed that you could model graph problems with quantum computing, I knew I had to do this.

At first, the code was somewhat unwieldy as I had never worked with a quantum computer before. But eventually I got around to understanding the code and then

combined my previous experience with pygraphviz with this program. In the future I hope to expand upon this and perhaps allow for more ways of input; currently the way graphs are inputted is relatively basic. I'm hoping to get full Graphviz/DOT import support as the pygraphviz module is extremely versatile. Not to mention visual learners like me understand the implications of graph problems a lot more when there's something tangible on the screen.

In my short dip into the quantum computer world, I learned how easy and accessible it was to access a quantum computer. Even though the DWave quantum computer can only deal with a small subset of problems to be solved (as it can only do quantum annealing), it opened my eyes to later prospects in the field. Who knows, maybe we will be able to feasibly run Shor's/Grover's algorithm and potentially beat AES-128. Or maybe quantum computing will just turn out to be a fad. I'm completely onboard now and hope to see major improvements someday.