Motivation: Quantum computing is a part of computer science that is very new, so not a lot is known about the field. It opens up the possibility to explore old problems in a completely new way. The NP class of problems – some of the most difficult problems in computing because of the way their runtimes grow combinitorially with the size of the data set – is fascinating to me because they present a true challenge of algorithmic efficiency where even small improvements on complexity can lead to saving a lot of time when sovling these problems on larger datasets. The P = NP problem is probably the most significant open question in computer science, so making any advancements in the subject would be very exciting.

Significance: NP problems are used in a variety of practical applicationns. Travelling salesman-type problems crop up in GPS navigation and routing. Integer programming is used in business to optimize [things].

Research Questions: An analysis of the practicality of a quantum computing algorithm for solving a NP-class problem versus its classical computing couterpart algorithm. [runtime on varying dataset sizes; cost of implementation; time of when we expect QC of such calibur to be available; since QCAs are by definition probabilistic, how does this factor into their useability and runtime].

Approach: Write (or find an existing) QCA for a given NP problem. Calculate its runtime on various dataset sizes. Compare results to the best known algorithm for solving the same problem, classically.

Anticipated Results: I expect QCs to be faster than classical computers through all sizes of datasets. Given that classical computers can solve NP problems, now, with dataset sizes in the thousands (10s of thousands) in a reasonable time, we probably won’t care about using QCs for such problems until we reach computers with hundreds (thousands) of qubits [current best is like 8 qubits].