Ethan Mayer (emayer4)
CSE 403 AA
Project Pitch
QuickUnit

JUnit, although very helpful and functional, has many problems that can make testing very slow. When testing, especially with unit tests, a large slew of tests are written, so as to hammer out the many edge cases and behavior domains of the program, making for long test runtimes. Often times, the tests that one cares about are near the end of the testing process, and is only discovered after waiting for almost all of the tests to finish.

A natural solution would be to order tests on their likelihood of failure, given that other tests are going to fail, that the tests that a developer most likely cares about will be ran first, avoiding the heavy downtime of large test runs. This is exactly what QuickUnit aims to achieve. The general approach would be to keep track of previous failures and successes of tests, and generate conditional probability (implicitly) by looking at prior beliefs of test outcomes given other test outcomes, and generate an ordered list, starting with the tests that are most likely to fail. This would effectively allow a developer to see which tests have failed first, instead of waiting to see which will fail, and which won't, saving tons of time on testing.

There are many challenges that accompany this, however, mainly revolving around having to keep track of many previous test runs, which could potentially be a large time and space waste. In the worst case, the computation time required to order the tests could outweigh the time the developer saves, making the tool pointless. To make this tool fast, it's critical that previous test runs and conditional probabilities are stored in an efficient manner.