

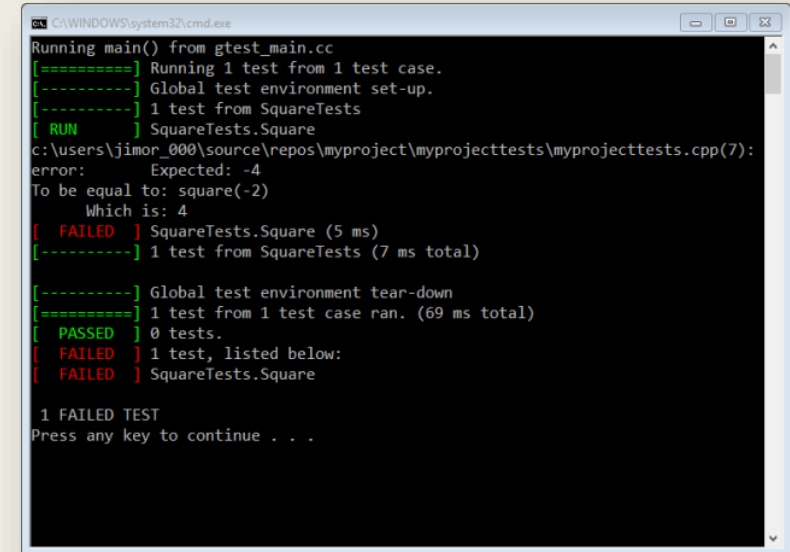


QUICKUNIT

Ethan Mayer (emayer4)

Motivation

- Testing usually entails enormous test sets
 - *Often can take hours, if not days, to run*
 - *Industry relies on testing to ensure quality*
- Tests might fail late, wasting a ton of time
- Tests often fail in tandem with other tests
 - *Many failed tests may overwhelm/confuse developer*
- Static ordering is the main problem
 - *No notion of tests being ordered by probability of failure (most important)*



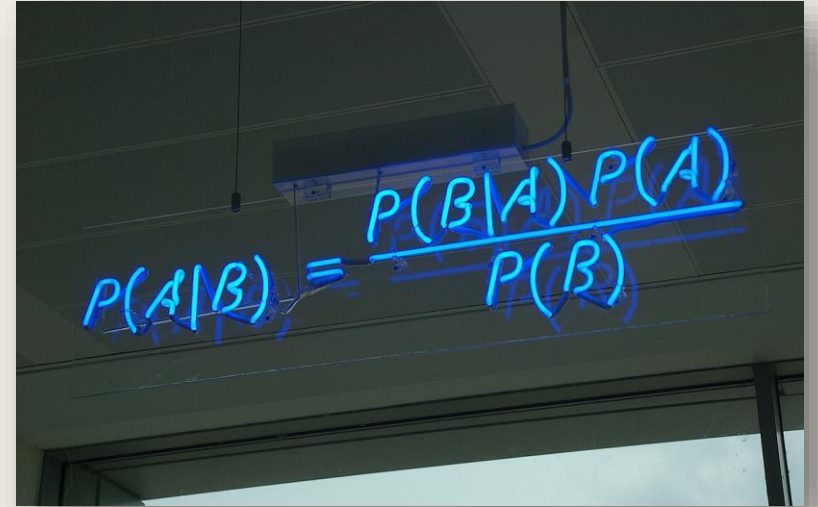
```
C:\WINDOWS\system32\cmd.exe
Running main() from gtest_main.cc
[=====] Running 1 test from 1 test case.
[-----] Global test environment set-up.
[-----] 1 test from SquareTests
[ RUN    ] SquareTests.Square
c:\users\jmor_000\source\repos\myproject\myprojecttests\myprojecttests.cpp(7):
error:         Expected: -4
To be equal to: square(-2)
             Which is: 4
[  FAILED ] SquareTests.Square (5 ms)
[-----] 1 test from SquareTests (7 ms total)

[-----] Global test environment tear-down
[=====] 1 test from 1 test case ran. (69 ms total)
[ PASSED ] 0 tests.
[  FAILED ] 1 test, listed below:
[  FAILED ] SquareTests.Square

1 FAILED TEST
Press any key to continue . . .
```

Approach & Challenges

- Build on top of JUnit
- Order tests based on their likelihood of failure
 - *Combination of prior failure rates*
 - *Account for changes the developer made, based on how similar changes affected tests*
 - *Allows dev to look at important tests first*
- Use Bayesian analysis to rank tests with better order, a test has failed
- Challenges revolve mainly around finding the likelihood of a certain test to fail, given that some other test has failed
 - *May require a machine learning model, or some learning algorithm*



A photograph of a whiteboard with the Bayesian formula $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$ written in blue marker. The whiteboard is mounted on a wall, and the background is dark.