CS 162 Lab 10 Reflection

Testing Results:

To begin with, I did not use a flag for compiler optimizations. I initially ran 10,000 tests of all four functions with the input value being 20. The resulting times were 0 seconds for every function except the double recursive Fibonacci function, which had results between 16 and 19 seconds.

Next, I increased the number of tests to 100,000 for each function. The resulting times were again 0 second for both factorial functions, 1 second for the iterative Fibonacci function, and 162 seconds for the double recursive Fibonacci function.

Last, I set the –O2 compiler optimization flag and ran 100,000 tests for each function. The results were the same for all functions except the double recursive one, which took about 90 seconds to complete all tests.

Testing Analysis:

Interestingly, the server and C++ seem to be so fast that even large numbers of tests hardly suggest any discernable difference between the three functions other than the double recursive function. I had expected the –O2 flag to only improve the tail recursion function, but unfortunately the ctime.h header doesn't provide much granularity of time measurement, so it's hard to say how much the optimization helps in that case. It would be a fun thing to explore using C++11's standard library functions, but as I have a lot of work to do on the final project I will have to wait on that for now. Even more surprisingly, the optimizations cut the time needed to run the double recursive function by almost half. I am not sure what particular optimization may have been done, but it was very noticeable.