CSC161 2011S *Imperative Problem Solving*

Class 26: Interactive Testing

Held: Tuesday, 8 March 2011

Summary: We do some more testing.

Related Pages:

- EBoard.
- Lab: Testing in C.

Notes:

- Tonight's reading on Design by Contract.
- Exam 2 distributed.

Overview:

- Reflection.
- Two stories.
- Better testing.

Reflection

What did you learn from the previous class?

- Writing correct code can be hard.
- Writing good tests can be hard.
- Write tests first! (Or write tests and document first.)
- What else?

Two stories

Printout-based Testing at Bell Labs

Doug McIlroy, a famous computer scientist, posted the following to *The Risks Digest: Forum on Risks to the Public in Computers and Related Systems*:

> Sometime around 1961, a customer of the Bell Labs computing center questioned a value returned by the sine routine. The cause was simple: a card had dropped out of the assembly language source. Bob Morris pinned down the exact date by checking the dutifully filed reversions tests for system builds. Each time test values of the sine routine (and the rest of the library) had been printed out. Essentially the acceptance criterion was that the printout was the right thickness; the important point was that the tests ran to conclusion, not that they gave right answers. The trouble persisted through several

deployed generations of the system.
McIlroy, Doug (2006). Trig routine risk: An Oldie. Risks Digest 24(49), December 2006.

If, instead of a thick printout, Bell Labs had arranged for a count of successes and a list of failures, they (and their customers) would have have been in much better shape.

Binary Search in Java

Josh Bloch, the software engineer responsible for implementing the binary search in the Java libraries, wrote the following:

> The version of binary search that I wrote for the JDK contained the same bug. It was reported to Sun recently when it broke someone's program, after lying in wait for nine years or so.
> ...
> The bug is in this line:
> `6: int mid =(low + high) / 2;`
> In *Programming Pearls* Bentley says that the analogous line "sets m to the average of l and u, truncated down to the nearest integer." On the face of it, this assertion might appear correct, but it fails for large values of the int variables low and high. Specifically, it fails if the sum of low and high is greater than the maximum positive int value (2^{31} - 1).
> Bloch, Josh (2006). Extra, Extra - Read All About It: Nearly All Binary Searches and Mergesorts are Broken. *Google Research Blog*, available at `http://googleresearch.blogspot.com/2006/06/extra-extra-read-all-about-it-nearly.html`. (Dated June 2, 2006.)

Systematic Testing

Have you identified (another) bug? There is one. How did I find it? Incredibly systematic testing.

- Try lots of sizes of arrays (say, 0 to 33).
- Try *every* position in each array.
