

CS 482, Spring 2009
Assignment #5
Testing

Due 4/22/09, in class

Part 1: Path Testing

For a function from one of your 301 or 302 programs (pick one that's between 30 and 50 source lines of code) create a statement and edge adequate test suite. For each test in the suite include a description of the test (i.e., a rationale), the process used to run the test, the input, and the expected output from the test.

What to hand in

- (1) The function you choose.
- (2) Its control-flow graph.
- (3) The test suite (input's should explicitly map variables to values).
- (4) The path taken by each test case. (Highlight each path on a single copy of the flow graph using different colors or different patterns as needed.)
- (5) An annotated script of the actual output for each test run.

Notes

- (1) Please type your answer (you may hand draw any figures).
- (2) Better answers minimize the size of the test set.

Part 2: Black Box Testing

Using the techniques from Chapter 8 perform thorough black-box testing of the sort method provided in the file `sort.class` available from the course web page. This file provides an API of three methods:

```
sort(int [] initialArray, int elementCount) -- the constructor
void print() -- print the current array
void sort() -- sort the current array
```

Here is an example use

```
public static void main(String [] args)
{
    int [] x = {1,6,4,2};
    sort t = new sort(x, 4);
    t.print();
    t.sort();
    t.print();
}
```

What to hand in

- (1) The test suite (annotate each test case with "passed or "failed").
- (2) A one page write up describing any faults found. Include with each your speculation as to the cause and a potential fix.
- (3) Data collected using different sized arrays. Use this data to predict the complexity of `sort.sort`. Don't hesitate to include a chart of the data :)

Notes

- (1) Please type your answer.