CS-315 Fall 2016
JWJ CS UK
Homework Assignment – Experimental Analysis

Due: solve before the Dead Week

Learning objectives:

- Experimental analysis of the running time;
- Measuring the clock time or/and counting the number of dominating operations for an algorithmic solution;
- Plotting graphs to visualize the program performance with gnuplot or with Excel;
- Implementing and comparing algorithms with respect to their running time;
- Designing inputs for program testing.
- Following provided examples: cs315002f16-running-time-example.tar

Specifications:

Your tasks are:

1. Read specs for problem UVA: 00567 - Risk.
2. Analyze the provided solution. See cs315002f16-running-time-example.tar for uva00567.cpp .
3. Design a plan for testing the running time: some changes in the program will be needed to accomodate larger than 25 test cases.
4. Instrument the program with calls to chrono functions and the counter to count the number of min operations.
5. State theoretical results for the total number of min operations: the number of comparisons between the current and candidate distance.
6. Design a number of test cases: input graphs of sizes, for example, $m = 10, 20, 30, 40, 50, 60, 70, 80, 90, 100$. Perform the experimental analysis of your program.

   Input has the format as described in UVA 567 - Risk problem. Input sizes (see above) used for testing may need adjustments to provide reasonable execution times; the time is measured with chrono library functions.
7. For each input of size m, tabulate the count (number) of min operations and the time to execute the program;
8. Graph count as a function of m; use gnuplot or Excel.

9. Graph the time as a function of m; use gnuplot or Excel.
10. Graph the ratio time/count as a function of m; use gnuplot or Excel.
11. Graph the ratio time/theoretical running time; use gnuplot or Excel.
12. Use instrumentation for your experimental analysis:
    - write a program or script to generate inputs for your testing;
    - Write scripts to organize your experiments, and
    - write script for GNUPLOT to visualize results (if gnuplot is used).
13. Discuss the results. In particular, do your experimental results (as illustrated in your graphs) agree with the theoretical running times for your programs?

How to turn the project in: Your homework (implementation, experiments) should be deployed in the Multilab; I will test your programs on one of the Multilab computers. The entire project should be submitted electronically by the midnight of the due day through the submission Web page.