Project 3

CS 3343 – Fall 2006 assigned November 6, 2006
Tom Bylander, Instructor due November 30, 2006

Your code and report must be submitted in a zip file to WebCT. All code and analysis should be your own. Late reports are accepted with 50% off. Waiting until the last minute and encountering a technical problem with your computer or WebCT means that your report will be late even if you emailed to me or placed it under my door. Plan accordingly.

The goal of this project is to perform an empirical analysis of shellsort (p. 164) using three different "increment sequences." In the insertion sort pseudocode (p. 161), j is set to $i-1$ and within the inner loop, j is decremented by 1. In shellsort, we repeatedly call insertion sort, but instead of changing j by 1, we change j by an "increment" h,[1] where h is a parameter that is changed for each call to the modified insertion sort. That is, j is set to $i-h$ and decremented by h in the inner loop. The value of h is chosen by an increment sequence, e.g.,

$$1, 4, 13, 40, \ldots \quad h_0 = 1, h_k = 3h_{k-1} + 1 \text{ for } k \geq 1$$
$$1, 8, 23, 77, \ldots \quad h_0 = 1, h_k = 4^k + 3(2^{k-1}) + 1 \text{ for } k \geq 1$$

The first value of h is the largest increment less than the number of elements to be sorted, then each smaller increment is used. Use the above two increment sequences plus a third one of your own invention.

I would prefer that your report be submitted as a PDF or Postscript file, or a group of linked HTML files. Submitting as a Word or OpenOffice document is ok though. I want to see the report as a single document. I do not want to open plots separately.

The input to your programs should be arrays of random numbers. Your code needs to count the number of basic operations (comparisons) and to record the time used to perform the sorting. To get reasonable averages for a given number of elements, you should run an algorithm several times for that number of elements, making sure you use different random values each time. You need to choose many different numbers of elements (at least 10, better to have 20).

Your code should also test that the algorithm is correct. For sorting, the numbers should be in ascending order and should be the same numbers you started with. Testing the algorithm should be excluded in the timings.

Your report should summarize your code: what files are included, what each method or function does, and how to invoke your program.

Your report should include two plots: the number of comparisons by the increment sequences, and the timings for the increment sequences. Do not produce different plots for different increment sequeces; it's very hard to compare across different plots. Each plot should include an n^α curve for each increment sequence generated from the data, where α is a exponent that seems to fit the data best. You can accomplish this by modifying the `QLS.java` code: change both occurences of "`x*x`" to "`Math.pow(x,`α`)`". A lower value of the `sse` variable indicates a better fit.

[1] It would be more accurate to call h a "decrement," but that's terminology for you.

Your report should describe each plot and the conclusions from the results. For example, which algorithm is better? What simple n^α function appears to correspond to the basic operation count and the timings? Does the basic operation count correspond to the timings?