CSU200 Discrete Structures	Dan Brown
Spring 2008	April 7, 2008

Written Homework 04

Assigned: Mon 07 Apr 2008
Due: Mon 14 Apr 2008

Instructions:

- The assignment is due at the *beginning* of class on the due date specified. Late assignments will be penalized 50%, as stated in the course information sheet. Late assignments *will not be accepted* after the solutions have been distributed.

Problem 1 [50 pts;, (20,15,15)]: Comparisons of Functions

In class, we discussed two quadratic sorting algorithms (Insertion-Sort and Selection-Sort) as well as one $n \lg n$ sorting algorithm (Merge-Sort). Dozens, if not hundreds, of other sorting algorithms have been developed. Another well-known sorting algorithm is Shell-Sort whose asymptotic running time is on the order of $n \lg^2 n$, when implemented appropriately.

In the problems that follow, you will compare these three algorithms for sorting. Ignoring lower order terms and constant factors, let $T_1(n)$, $T_2(n)$, and $T_3(n)$ be the "effort" required by Insertion/Selection-Sort, Shell-sort, and Merge-Sort, respectively, to sort a list of length n. We have

$$\begin{aligned} T_1(n) &= n^2 \\ T_2(n) &= n \lg^2 n \\ T_3(n) &= n \lg n \end{aligned}$$

where $\lg n$ is $\log_2(n)$.

i. On a *single* sheet of graph paper, plot the effort required by each of the these algorithms when run on lists of length $n = 2$, 4, 8, 16, and 32. For each algorithm, connect the plot points with a smooth, hand-drawn curve. See the plots given in the "Exponentials and Logs" appendix of the text for examples of what youshould do. You may print a piece of graph paper from the PDF located at the following URL:

`http://www.printfreegraphpaper.com/`

(If you view this assignment on-line, you may simply click on the above hyperlink.)

ii. Suppose that you were given a budget of 1,000 units of "effort." For each of the three algorithms, determine the largest list length such that the sorting effort required is guaranteed to be at most 1,000.

iii. How many times larger is the list that MERGE-SORT can handle, as compared to the lists that INSERTION/SELECTION-SORT and SHELL-SORT can handle? How many times larger is the list that SHELL-SORT can handle, as compared to the list that INSERTION/SELECTION-SORT can handle?

Problem 2 [50 pts, (25,25)]: Recurrences

In each of the following problems, solve the recurrence using the method described in class and the text. *You must show your work.*

Assume a base case of $T(1) = 1$. As part of your solution, you will need to establish a pattern for what the recurrence looks like after the k-th iteration. For this assignment, you need *not* formally prove that your patterns are correct via induction, though you will lose points if your patterns are not correct. Your solutions may involve n raised to a power and/or logarithms of n. For example, a solution of the form $8^{\log_2 n}$ is unacceptable; this should be simplified as $n^{\log_2 8} = n^3$.

i. $T(n) = 9\,T(n/3) + n^2$.

ii. $T(n) = 9\,T(n/3) + n$.