Homework 01

Assigned: Thu 15 Sep 2005
Due: Wed 21 Sep 2005

Instructions:

In the problems that follow, you will consider three of the algorithms for search which we discussed in class: Ordered-Linear-Search, Chunk-Search, and Binary-Search. Let $T_1(n)$, $T_2(n)$, and $T_3(n)$, respectively, be the number of comparisons[1] required by these algorithms when run on a list whose length is n. We have

$$\begin{aligned} T_1(n) &= 2n \\ T_2(n) &= 2\sqrt{2n} \\ T_3(n) &= 2\log_2(n). \end{aligned}$$

In the problems that follow, you will compare and contrast the growth rate of these functions.

Problem 1 [20 pts]**:** On a *single* sheet of graph paper, plot the number of comparisons required for each of the three algorithms when run on lists of length n = 1, 2, 4, 8, and 16. For each algorithm, connect the plot points with a smooth, hand-drawn curve. See the plots given in the "Exponentials and Logs" handout for examples of what you should do. You may print a piece of graph paper from the PDF located at the following URL:

`http://www.printfreegraphpaper.com/gp/c-i-14.pdf`

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

Problem 2 [24 pts, (12,12)]**:**

i. Suppose that you were given a budget of 20 comparisons. For each of the three algorithms, determine the largest array length such that the number of comparisons made is guaranteed to be at most 20.

ii. How many times larger is the array that Binary-Search can handle, as compared to the arrays that Chunk-Search and Ordered-Linear-Search can handle? How many times larger is the array that Chunk-Search can handle, as compared to the array that Ordered-Linear-Search can handle?

Problem 3 [28 pts, (12,4,12)]**:** Moe, Larry, and Curly have just purchased three new computers. Moe's computer is 10 times faster than Larry's and 50 times faster than Curly's.[2] However, Moe runs Ordered-Linear-Search on his computer, while Larry and Curly run Chunk-Search and Binary-Search, respectively. Moe, Larry, and Curly begin to perform searches over various data stored on their computers. . .

[1] worst-case. . .

[2] In other words, Moe's computer can perform 50 comparisons in the time it takes Curly's computer to perform one comparison, and Moe's computer can perform 10 comparisons in the time it takes Larry's computer to perform one comparison.

i. How large must n (the size of the array) be so that Curly's computer begins to outperform Moe's?

ii. How large must n be so that Larry's computer begins to outperform Moe's?

iii. How large must n be so that Curly's computer begins to outperform Larry's?

Hint: For parts of this problem, you will have to solve an equation that involve n (or $\sqrt{n}$) and $\log_2 n$. Such equations cannot typically be solved *analytically*, i.e., by applying the rules of algebra to obtain a formula for n. Such equations are often solved *numerically* by using... binary search! Consider the equation

$$10 \log_2 n = n.$$

First, find two initial values of n, one of which causes the left-hand side of the equation to exceed the right, and the other of which causes the right-hand side of the equation to exceed the left. The solution to the equation lies somewhere between these two values. For example, when $n = 2$, the left-hand side of the equation is 10 while the right-hand side is 2. Conversely, when $n = 128$, the left-hand side is 70 while the right-hand side is 128. The solution to this equation lies somewhere between $n = 2$ and $n = 128$. One could then apply binary search in this range to find the solution in question. (Think about which half of the interval one should search and why.)

Problem 4 [28 pts, (14,14)]**:** Consider the setup described in Problem 3 above.

i. Moe and Curly both run a search on the same data set. Despite the fact that Curly's machine is 50 times slower than Moe's, Curly's machine performs the search 100 times faster than Moe's machine. How large is the data set?

ii. Suppose that Moe switches to CHUNK-SEARCH. On this same data set, will Moe's machine now outperform Curly's? Explain.

Hint: Again, for part of this problem, you will need to solve an equation using binary search.