Data Structures: cs271
Denison University
Professor Thomas C. Bressoud

Jan. 18, 2012
Homework Set 1

Homework Set 1

This problem set is due **Friday January 27** at **8:30AM**.
Solutions should be turned in in PDF form using LaTeX.
A template for writing up solutions in LaTeX is available on the course website.
Remember, your goal is to communicate. Full credit will be given only to the correct solution which is described clearly. Convoluted and obtuse descriptions might receive low marks, even when they are correct. Also, aim for concise solutions, as it will save you time spent on write-ups, and also help you conceptualize the key idea of the problem. Demonstration of your work/reasoning can result in partial credit.

---

1. **(3 points)** Suppose we are comparing implementations of insertion sort and merge sort on the *same* machine. For inputs of size n, assume that insertion sort runs in $4n^2$ steps, while merge sort runs in $400n \log_2 n$ steps. For which values of n does insertion sort beat merge sort?

2. **(10 points)** Prove by induction the following theorem:

$$\text{If } N \geq 1 \text{ then } \sum_{i=1}^{N} i^2 = \frac{N(N+1)(2N+1)}{6}$$

3. **(14 points)** For each running time function $f(n)$ and time t in the following table, determine the largest size of n of a problem that can be solved in time t by an algorithm that takes $f(n)$ microseconds to solve the problem.

	1 second	1 hour	1 day	1 year	1 century
$\log_2 n$					
$\sqrt{n}$					
n					
$n \log_2 n$					
n^2					
2^n					
$n!$					

4. **(15 points)** Consider the searching problem:

   Input: a sequence $A = \langle a_1, a_2, \ldots, a_n \rangle$ and a value v.

   Output: an index i such that $v = A[i]$ or $\perp$ if v is not an element of A.

   The *linear search* algorithm scans through the sequence one element at a time until an element $A[i] = v$ is found, at which point if stops and returns i.

   (a) Write pseudocode for linear search.

   (b) Using a loop invariant, prove the algorithm is correct. Make sure you argue all three necessary properties of the loop invariant.

   (c) What are the average case and worst case running times of linear search in Θ notation. Justify your answer. For the average case, assume v is equally likely to match any element in the array.

5. **(10 points)** Hybrid Sort

   Although merge sort runs in $\Theta(n \lg n)$ worst-case time and insertion sort runs in $\Theta(n^2)$ worst-case time, the constant factors in insertion sort can make it faster in practice for small problem sizes on many machines. Thus, it makes sense to *coarsen* the leaves of the recursion by using insertion sort within merge sort when subproblems become sufficiently small. Consider a modification to merge sort in which n/k sublists of length k are sorted using insertion sort and then merged using the standard merging mechanism, where k is a value to be determined.

   (a) Show that insertion sort can sort the n/k sublists, each of length k, in $\Theta(nk)$ worst-case time.

   (b) Show how to merge the sublists in $\Theta(n \ln(n/k))$ worst case time.

   (c) What is the worst-case running time for the hybrid algorithm?

6. **(20 points)** Perform an experimental analysis of insertion sort, merge sort, and the above hybrid sort, as we scale the input size n. Your professor will supply much of the surrounding code, so that you may focus on the implementation of the three sorts and on their analysis. You may use Excel or OpenOffice to generate the figures, but these should result in figures that get included into your LaTeXhanding, and you should give a description of the relative performance and how you arrived at the k for the merge sort.

   You should also hand in a tar, complete with Makefile, for the instructor to verify correctness against input data sets.