CPS130: Homework 2
Due: Wednesday, September 28, 2005

1. Although Merge sort runs in $\Theta(n \log n)$ worst-case time, the constant factors allow Insertion sort (which runs in $\Theta(n^2)$ time) to actually run faster for small values of n. Since Insertion sort is faster, it makes sense to use it when Merge sort has reduced the list to a certain size. Consider a modification of merge sort in which n/k sublists of length k are sorted using insertion sort, then merged using standard merging. k is just a number; we'll determine its value later.

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

   b) Show that you can merge all of the sublists in $\Theta(n \log(n/k))$ time.

   c) Given that this new sorting method runs in $\Theta(nk + n \log(n/k))$ time, what is the largest (asymptotic) value of k that makes this algorithm have the same running time as merge sort? Express your answer in Θ notation.

   d) How should we choose k in practice?

2. Given two sorted lists A and B of n elements each, describe an efficient algorithm to find the median of both lists - ie, the median if the two lists were to be combined. This can be done in $\Theta(\log n)$ time.

3a. Describe an algorithm (pseudo code or written description) to find the second smallest element of a list of n elements with $n + \lceil \log n \rceil - 2$ comparisons.

3b. Can we adapt the same approach to find the third smallest? Explain.