Exercises for Lecture 2

March 1, 2023

Exercise 1 (Cormen's Problem 2-1). Although merge sort runs in $\Theta(n \log n)$ 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 reduce the number of levels 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.

1. Show that insertion sort can sort the n/k sublists, each of length k, in $\Theta(kn)$ worst-case total time (that is: the time required to sort all of them).

2. Show how to merge the sublists in $\Theta(n \log(k/n))$ worst-case total time (that is: the time required to merge all of them).

3. Given that the modified algorithm runs in $\Theta(nk + n \log(n/k))$ worst-case time, what is the largest value of k as a function of n for which the modified algorithm has the same running time as standard merge sort, in terms of Θ-notation?

4. How should we choose k in practice?

Exercise 2 (Cormen 6.1-5). Is an array that is in sorted order a min-heap?

Exercise 3 (Cormen 6.1-6). Is the array $A = [3, 17, 14, 6, 13, 10, 1, 5, 7, 12]$ a max-heap?