CS 374 Lab 10: Merge Sort and Recurrences

Date: February 24, 2016.

Problem 1. [Category: Design] Given a sequence $a_1, a_2, \ldots, a_n$ of n distinct numbers, an *inversion* is a pair $i < j$ such that $a_i > a_j$. Note that a sequence has no inversions if and only if it is sorted in ascending order.

1. Adapt the merge sort algorithm to count the number of inversions in a given sequence in $O(n \log n)$ time. You can find the detailed description of this in the Kleinberg-Tardos book (Chapter 5). *Hint:* Modify the algorithm for Merge Sort.
2. **Think about at home.** Call a pair $i < j$ a *significant* inversion if $a_i > 2a_j$. Describe an $O(n \log n)$ time algorithm to count the number of significant inversions in a given sequence.

Problem 2. [Category: Analysis] Give asymptotically tight solutions to the following recurrences. For the third problem *prove* your upper bound via induction.

1. $T(n) = T(\sqrt{n}) + \log n$ for $n \geq 4$ and $T(n) = 1$ for $1 \leq n < 4$.
2. $T(n) = T(n/5) + T(n/10) + T(7n/10) + n$ for $n \geq 20$ and $T(n) = 1$ for $1 \leq n < 20$.
3. $T(n) = T(n/6) + T(n/10) + T(7n/10) + n$ for $n \geq 20$ and $T(n) = 1$ for $1 \leq n < 20$.