Homework 2
Due 9/22/16 (Thursday)

1. *Recurrence relations.* Problem 2.5 (b to e only) and Problem 2.12.

2. *Inversions.* Assume the array $A[1 \ldots n]$ simply contains the numbers 1 through $n$. We shall say that it has an *inversion* if there is a pair of indices $i$ and $j$ such that $i < j$ but $A[i] > A[j]$. For example, if the numbers in $A[1 \ldots n]$ are arranged from smallest to largest, $A$ has no inversions. If, on the other hand, the numbers are ordered from largest to smallest, $A$ has $\binom{n}{2}$ inversions.

   a. Given $A[1 \ldots n]$, describe a divide and conquer algorithm that counts the number of inversions in the array. (Hint: Modify mergesort.) What is the running time of your algorithm?

   b. Let us say that an inversion is *significant* if $i < j$ and $A[i] > 2A[j]$. Again, describe a divide and conquer algorithm that counts the number of significant inversions in the array. What is its running time?

3. *Constructing skylines.* Suppose we are given the locations and sizes of a city’s box-like buildings. That is, for every building $b$, we have $(x_\ell(b), x_r(b), h(b))$ where $(x_\ell(b), x_r(b))$ denotes its location on the $x$-axis and $h_b$ denotes its height. Visually, here’s how the 5 buildings $(2, 5, 8), (4, 7, 3), (8, 11, 4), (9, 13, 2), (10, 14, 7)$ look:

   ![Diagram of buildings]

   Here’s their skyline:

   ![Diagram of skyline]
Notice that the buildings’ skyline can be specified by the 4 horizontal line segments that make up the scene. Thus, if we express each horizontal line segment \( s \) as \((x_\ell(s), x_r(s), h(s))\) where \((x_\ell(s), h(s))\) is its left endpoint and \((x_r(s), h(s))\) is its right endpoint, we can now describe the skyline as \((2, 5, 8), (5, 7, 3), (8, 10, 4), (10, 14, 7)\).

a. Given information for \( n \) buildings \((x_\ell(1), x_r(1), h(1)), \ldots, (x_\ell(n), x_r(n), h(n))\), design a divide and conquer algorithm that outputs its skyline. (Hint: Think about how you would merge two skylines.)

b. Briefly argue why your algorithm works and determine its running time.