Please do all problems; we will grade a subset of the assigned problems (same subset for everyone). Please follow all of the instructions given in the handout for Homework 1.

1. (12 points) Ex. 13.2-4, p. 314. A careful, well-worded answer should be given. Pseudocode is not required.

2. (12 points) Ex. 13.3-5, p. 322. A careful proof is expected. Consider using induction on $n$ and examining the various cases that can arise in RB-INSERT-FIXUP. (Figures 13.5 and 13.6 are helpful.)

3. (16 points) Problem 13-2, p. 332-333. (Skip part (e) as it is symmetric to part (b).) Supplement your answer with short code fragments, as appropriate.

4. (10 points) Ex. 14.3-3, p. 353. Describe the main ideas underlying your solution (from which correctness should be evident), give pseudocode, and analyze the running time.

5. (12 points) Let $T$ be a red-black tree with $n$ nodes, where each node has the usual fields: key, color, and pointers to its parent and children. In addition, each internal node $x$ has an auxiliary field, $\text{diff}$, whose value is the difference between $x$’s key and the minimum key in the subtree rooted at $x$ (the subtree includes $x$). The auxiliary field is not of interest for external nodes. Argue carefully, using the General Augmentation Theorem (GAT), that the auxiliary field can be maintained during insertions and deletions without affecting the $O(\log n)$ time bounds for these operations. (You cannot store any other information in the nodes.) You must use the GAT to make your argument, rather than trying to argue from first principles. The latter approach will earn no credit.

6. (14 points) Let $S$ be a set of $n$ line segments in the plane, where each segment is either horizontal or vertical and is specified by the coordinates of its two endpoints. (Assume, for convenience, that no two endpoints in the input have the same $x$- or $y$-coordinate.) Give an $O(n \log n)$-time sweepline algorithm, to count the number of pairs of horizontal-vertical segments that intersect. The output of your algorithm should merely be an integer equal to the number of intersecting horizontal-vertical pairs. For instance, for the example given in the figure on the next page, the output should be “5”, corresponding to the intersecting pairs $(a, h)$, $(b, h)$, $(c, j)$, and $(d, j)$. (We are not interested here in knowing which pairs of segments intersect, just the number of such pairs.) Do the following: (a) Describe the main ideas behind your solution (from which correctness should be evident), (b) give pseudocode, and (c) analyze the running time. You may use any data structure discussed in class as a black-box, i.e., you do not have to write code for the operations you do on this structure (as long as you do not modify the structure).