

Today's topics: Red-black trees; simple sorting. Textbook sections: Chapter 19.5; 8.1–8.3.

Next class: AA-trees; Textbook §§ 19.6, AA-trees paper.¹

Reminder: Read material *before and after* class.

1. List the members of your group below. Underline your name.

2. Describe, in your own words, the *essence* of the *top-down* insertion algorithm for *red-black trees* (approx. 100 words). Depict the red-black tree resulting from the sequential insertion of

1, 2, 3, . . . , 10, 20, 19, . . . , 11

into an empty tree, using top-down insertion. All intermediate trees need not be depicted, but it is advisable to depict at least a few.

¹Arne Andersson, “Balanced Search Trees Made Simple,” in *Proceedings of the Workshop on Algorithms and Data Structures* (Montreal, Canada, 1993).

[additional space for answering the earlier question]

3. Delete each of the following keys (in given order) from the final tree of Question 2. Depict the state of the tree before and after each transformation required by the deletions.

9, 3, 1

4. Provide succinct Java code for *insertion sorting* an array of integers. Explain briefly why your code is correct.