Assignment 4, CS 251, Feb 19, 2019.

Exercise 1. LEAST COMMON ANCESTOR IN RED-BLACK TREES. Write an algorithm that, given pointers to nodes u and v in a red-black tree, finds the least common ancestor in $O(\log n)$ time.

Exercise 2. COMPLETE BINARY TREES AND RED-BLACK TREES. We are given a sorted array A with elements denoted by $A[1], \ldots, A[n]$.

(i) Write a linear time algorithm for making a complete binary search tree with these elements, where the complete tree is stored in the standard fashion in another array, B, i.e., $B[1]$ is the root, $B[2]$ and $B[3]$ are the children of the root, and so forth.

(ii) Write a linear time algorithm for making a complete binary search tree with these elements, where the complete tree is stored with pointers only.

(iii) Given the tree output in part (ii), fill in the color fields in all nodes to make it a proper red-black tree. (And this serves also as a proof that a complete binary tree can always be made into a red-black tree).

Exercise 3. SWEEPLINE ALGORITHMS. We are given n axis-aligned rectangles in the plane where each rectangle is of the form $[a, b] \times [c, d]$. Imagine that there is a light source in the northern direction, infinitely far away. Write an $O(n \log n)$ sweepline algorithm for finding all the rectangles that are partially illuminated by that light source.

Exercise 4. THE SMALLEST LARGER INTEGER DATA STRUCTURE. Design an efficient data structure for the following abstract data type. The object is a subset S of $\{1, 2, \ldots, n\}$, where n is fixed and known, and S is dynamic. The operations are as follows.

(i) ADD(i, S): adds i to S.

(ii) DELETE(i, S): deletes i from S.

(iii) MEMBER(i, S): returns true if $i \in S$.

(iv) NEAREST(i, S): returns the smallest $j \in S$ with $j > i$. It returns nil if such an integer does not exist.