ADS 2021: Week 10 Exercises

Exercises for week 10 of Algorithms and Data Structures at ITU. The exercises are from *Algorithms, 4th Edition* by Robert Sedgewick and Kevin Wayne unless otherwise specified. Color-coding of difficulty level and alterations to the exercises (if any) are made by the teachers of the ADS course at ITU.

3.2.9 - Green Draw all the different BST shapes that can result when N keys are inserted into an initially empty tree, for $N = 2$, 3 and 4.

3.3.2 - Green Draw the 2-3 tree that results when you insert the keys Y L P M X H C R A E S in that order into an initially empty tree.

3.3.3 - Green Find another insertion order for the keys S E A R C H X M that also lead to a 2-3 tree of height 1.

3.3.11 - Green Draw the red-black BST that results when you insert items with the keys Y L P M X H C R A E S in that order into an initially empty tree.

3.2.11 - Yellow How many binary tree shapes of N nodes are there with height N-1? How many different ways are there to insert N distinct keys into an initially empty BST that result in a tree of height N-1?

3.2.20 - Yellow Prove that the running time of the two-argument keys(lo, hi) in a BST with N nodes is at most proportional to the tree height plus the number of keys in the range.

Counting keys - Red Show how to augment a BST/red-black/2-3 tree such that you can perform counting queries in time proportional only to the tree height (and not the number of keys). As an example: Given a tree T and an interval [a,b], count the number of keys stored that fall into this range, i.e. $|\{k \in T | k \in [a, b]\}|$.

Counting odd keys - Red Assume the keys are integers. How can you support reporting the amount of odd keys in the interval, also in time proportional to the tree height?