
2. Determine the minimum number and the maximum number of red nodes in a red-black tree with 10 keys (internal nodes).
   For extra points argue why the numbers you have obtained are the extreme.

3. **AVL trees.** Assume that within each node of a binary search tree we store also the height of the pending subtree, i.e. the length of the longest descendant chain from this node).
   Consider trees which are almost balanced, i.e. they have the property that for every node, the height of the left subtree and the height of the right subtree differ by at most one. Empty tree (nil) has height 0.
   Use rotations to implement the **INSERT** operation in time proportional to the height of the whole tree. In particular the height records should contain correct values, as well as the resulting tree should be balanced.
   You may write a pseudocode, however a verbal description would be sufficient.

4. Design the optimal binary search tree for the six most frequent English words. Use internet to find these words and their frequencies (e.g. Brown corpus). Give credit to the source used.
   You may use computer for routine computations, but you should present the obtained values in a suitable way to argue that the resulting tree is optimal.