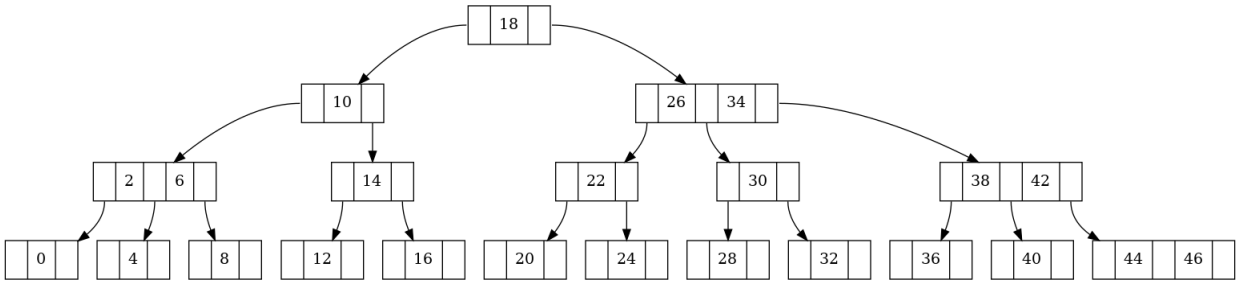
Tutorial 5

Data Structures 1, 21. 3. 2025 https://iuuk.mff.cuni.cz/~chmel/2425/ds1en/

Exercise 1 ((a,b)-trees in your hands)
In the figure, you can see a $(2,3)$-tree. Perform the following operations on this tree (always just one and then start with a new tree): Insert(7), Insert(48), Delete(44), Delete(40), Delete(32), Delete(30), Delete(16).



Exercise 2 (Correct parameter choice)
From the lecture, we know that an arbitrary sequence of m operations Insert and Delete on an originally empty $(a,2a)$-tree makes only $\mathcal{O}(m)$ node modifications (splits and merges). Show that this is not true for $(a,2a-1)$-trees. That is, for any m,n, design a sequence of m operations on a tree with $\Theta(n)$ nodes that changes $\Omega(m\log n)$ nodes in total.
You can start with $(2,3)$-trees and then generalize for an arbitrary a. You can also start with an arbitrary (valid) n-node tree and then show that you can indeed build the tree from the empty tree.

Exercise 3 (Better fill factor)
Try to tweak the (a,b)-tree and its operations Insert, Delete so that all its nodes can be a little more full. In particular, we want to be able to build a $\left(\frac{2}{3}b,b\right)$-tree (for b a multiple of three).

Exercise 4 ((a,b)-Join)
Design the Join operation for $(a,2a)$-trees: you have two $(a,2a)$-trees T_1,T_2 such that all keys in T_1 are smaller than all keys in T_2, and your goal is to build a single tree from the two trees. Beware that the trees can have different heights.
(For the analysis of Split, it might be useful to analyse the complexity more precisely than just $\mathcal{O}(\log n)$.)

Bonus exercises

Exercise 5 ((a,b)-Split)
Design the Split operation: you have an $(a,2a)$-tree T and a key k, and you want to split T into two trees such that one contains all nodes less than k and the other contains the rest. Moreover, we want the operation to run in time $\mathcal{O}(\log n)$.