Lecture At the lecture we will talk about two different kind of balanced search trees: red-black trees and 2-3-4 trees. You should read page 572–585 in "Algorithms in Java" by Sedgewick (can be found on DTU Learn).

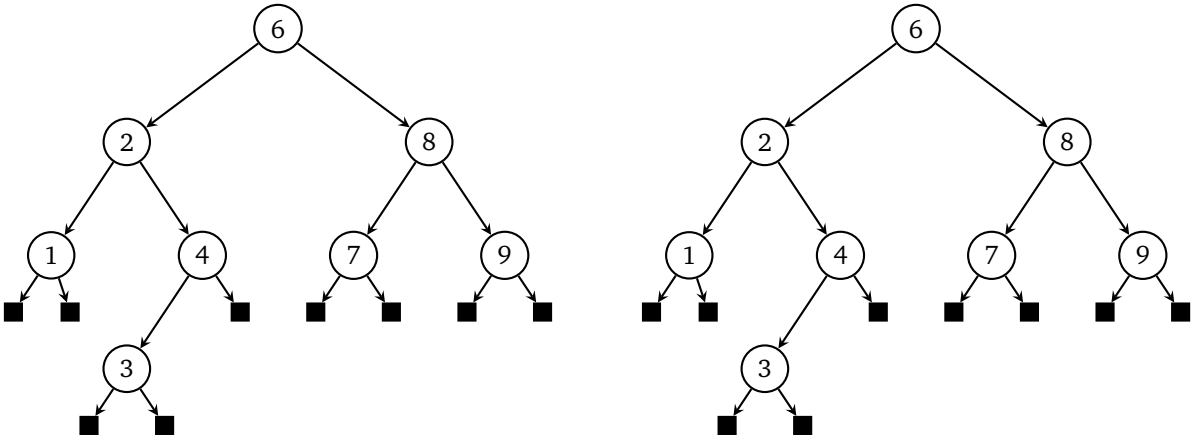
If you don't remember binary search trees, you should also read CLRS chapter 12 (*before* the lecture).

Exercises

1 Insertions in balanced search trees (w) Show the red-black tree and the 2-3-4 tree that results from inserting the keys 41, 38, 31, 12, 19 , 8 in this order into an initially empty tree.

2 Deletions in 2-3-4 trees (w) Show the trees that results from successively deleting the keys 8, 12, 19 , 31, 38, 41 in that order from the 2-3-4 tree in the first exercise ("Insertions in balanced search trees"). You should show how the tree look after each deletion.

3 Red-black trees Give two different colorings of the tree below that makes it a legal red-black tree.



4 Range queries Explain how to report all items with a key between a and b (both included) in a balanced binary search tree in $O(\log n + k)$ time, where k is the number of items that are returned.

5 Counting range queries Explain how to modify a balanced binary search tree to make it possible to report the number of items with a key between a and b (both included) in $O(\log n)$ time.

6 Databases You are working as a consultant for the company "*Boxes, Boxes and Boxes*", that sells boxes. They want a database containing information about all their boxes. Each box has an id, a size, a type, and a price. They want to be able to update the database with insertions and deletions of boxes. The database should support the following updates and queries efficiently:

- Insert(i, s, t, p): Insert a box with id i, size s, type t and price p into the database.
- Delete(i): Delete the box with id i from the database.
- Report-Price(a, b): Return the id of all boxes with a price between a and b.
- Find-Size(s): Return the id of the box with size closest to s.

6.1 Give a data structure supporting the required updates and queries. Analyse the space and the update and query times of your data structure.
You may assume that the prices and sizes of the boxes are unique.

6.2 Change your solution to handle the case where the sizes and prices are not unique.

7 Height of 2-3-4 trees Show that the height of a 2-3-4 tree has height at most $\lg(n+1)$, where n is the number of elements in the tree (not the number of nodes). *Hint:* First show by induction on h that $n \geq 2^{h+1} - 1$, where h is the height of the tree.

8 Deletions in 2-3-4 trees When we delete a node x in a 2-3-4 tree we do as follows:

Algorithm 1: Delete(T, x)

Search for x while maintaining the invariant that the current node not is a 2-node. (if current node is a 2-node split).
if *x is in a leaf* **then**
  Delete x.
else
  Find the successor y of x while maintaining the invariant that the current node not is a 2-node.
  Delete y from the leaf containing it.
  Replace x with y.
end

Argue that if x is not a leaf, then its successor y is always a leaf in a subtree rooted at a child of x.

9 Properties of red-black trees For the following statements give either a justification for the correctness of the statement or a counterexample.

1. A subtree of a red-black tree is itself a red-black tree.
2. Show that the *longest* (simple) path from a node x in a red-black tree to a descendant leaf has length at most $2\ell + 1$, where ℓ is the length *shortest* path from x to a descendant leaf.
3. What happens if we when inserting a new node connect it with a black edge instead of a red?

Puzzle of the week: The Blind Man A blind man was handed a deck of 52 cards with exactly 10 cards facing up. How can he divide it into two piles, each of which having the same number of cards facing up?