Lecture We will talk about amortised analysis and see a another kind of binary search trees, called splay trees.

You should read about

- amortised analysis in either Lecture 15 in the notes by Jeff Ericsson or CLRS chapter 17.
- Splay trees in section 16.5-16.6 in the notes by Jeff Ericsson.

The notes by Jeff Ericsson can be found here: http://www.cs.illinois.edu/∼jeffe/teaching/algorithms (also available on CampusNet).

Exercises

The difficulty of the exercise may be indicated with (w) and (*), where (w) means warmup exercise and (*) is a challenging exercise.

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.

Height of 2-3-4 trees Show that the height of a 2-3-4 tree has height at most $\lg(n+1)$.

Hint First show by induction on h that $n \geq 2^{h+1} - 1$, where h is the height of the tree.

Properties 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 (except the root might be red).
2. The sibling of a leaf node is either a leaf or red.
3. Show that the *longest* simple path from a node x in a red-black tree to a descendant leaf has length at most twice that of the *shortest* path from x to a descendant leaf.
4. Solve CLRS 13.1-4, 13.2-3, 13.3-1.
5. Show that after the rotations on Figure 13.5 and 13.6 in CLRS property 5 of a red-black tree is preserved.

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.

Exercise 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.

Exercise 2(*) Change your solution to handle the case where the sizes and prices are not unique.

Deletions in 2-3-4 trees When we delete a node x in a 2-3-4 tree we do as follows:
Delete(T, x):

1. Search for x while maintaining the invariant that the current node not is a 2-node. (if current node is a 2-node split it as in Figure 1).
2. If x is in a leaf: delete x.
3. If x is not in a leaf, replace it with its successor y. Let T' be the subtree rooted in a child of x that contains y. Recursively call Delete(T', y)

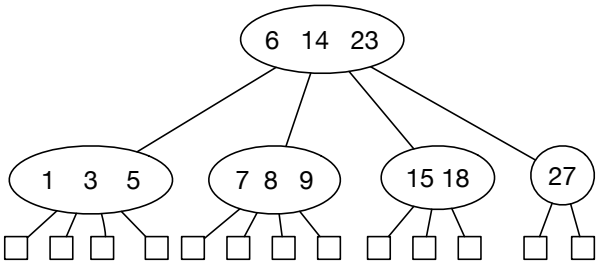
Argue that if x is not a leaf, then its successor y is always a leaf in a subtree rooted at a child of x.

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

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?

Mandatory assignment: Search trees

Question M.1 Let T be the 2-3-4 tree below. Perform the following operations on T: `Insert`(12), `Insert`(4), `Insert`(24), `Delete`(14), `Delete`(1). Show how the tree looks after each operation.



Question M.2 Draw the complete binary search tree of height 3 on the keys $\{0, \ldots, 14\}$ and add the black NIL leaves. Color the rest of the nodes in 3 different ways such that the black-heights of the resulting red-black trees are 2, 3, and 4.