# Homework Assignment 2

**Due: Friday, November 20**

Q1: $(a,b)$**–trees**

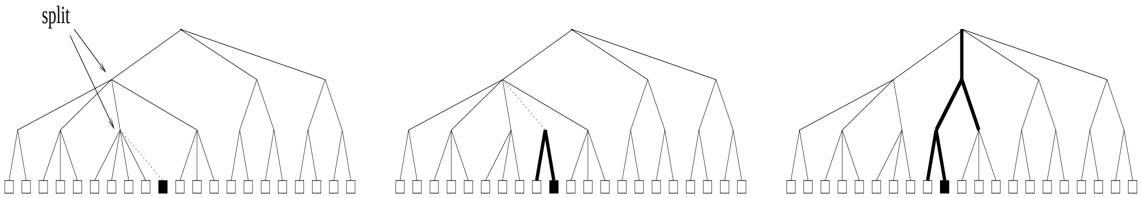
One class of balanced search trees are (2,4)–trees. A (2,4)-tree has the following structure

(a) All leaves have the same depth.

(b) All internal nodes $v$, satisfy $2 \leq \text{degree}(v) \leq 4$, where degree=#children.

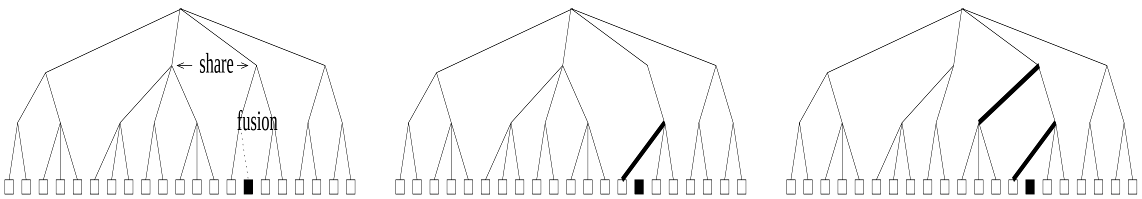
Elements are stored at the leaves, and each internal node of degree $k$ stores $k-1$ search keys to guide searches. By (b) a (2,4)–tree has height $O(\log n)$.

Insertion of a new element is done in three steps: 1) locate where to create a new leaf, 2) create the new leaf, and 3) rebalance the tree such that (a) and (b) are satisfied by splitting nodes of degree 5 into two nodes of degree 2 and 3.




Insertion.

Deleting an element: 1) locate the leaf storing the element, 2) delete the leaf, and 3) rebalance the tree such that (a) and (b) are satisfied by eliminating nodes of degree 1. Let $v$ be a node of degree 1. If $v$ is the root, delete $v$. If $v$ has a sibling of degree 2, fusion $v$ and its sibling. If $v$ has a sibling $w$ of degree $> 2$, it is sufficient to move one child of $w$ to $v$.




Deletion.

– Show that the amortized number of node splittings, fusions, and sharings is $O(1)$ for insert and delete, *i.e.*, that the amortized *restructuring cost* of (2,4)–trees is constant.

$(a,b)$-trees are a generalization of (2,4)-trees where each internal node satisfies $a \leq \text{degree}(v) \leq b$, except for the root that satisfies $2 \leq \text{degree}(v) \leq b$, where $b \geq 2a-1$ and $a \geq 2$.

– Show how to restructure $(a,b)$-trees after leaf insertions and deletions such that each operation moves amortized $O(1)$ edges between nodes, when $b \geq (2+\epsilon)a$, for some constant $\epsilon > 0$. Note that splitting a node moves $\Theta(b)$ edges between two nodes.

– What if $b = 2a$ (also known as B-trees)?

**Q2: The Stable Marriage Problem**

In a group of $n$ boys and $n$ girls, each girl ranks the $n$ boys according to her preference, and each boy ranks the $n$ girls according to his preference. A *marriage* is a matching between the boys and the girls such that everybody is married (a *perfect* matching in graph terminology). A marriage is *unstable* if there is a pair who are not married to each other but who like each other more than they like their respective spouses, otherwise it is *stable*. Prove that a stable marriage always exists and give an efficient algorithm to find one.