Splay tree properties

① as a BST

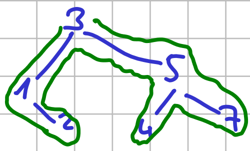
for m Splays on a n-node tree:

$O((n+m)\log n)$.

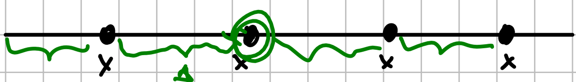
for m Find/Ins/Del on initially empty tree:

$O(m \log n)$.

② Sequential access of all items in increasing order takes $O(n)$ time.



⑤ Working Set Theorem



working set of ◎ := set of all items accessed since previous access to x (or from the beginning of time)

Thm: Given a Splay tree on an n-elem. set & seq. of accesses x_1 — x_m with working set sizes z_1 — z_m, total cost of accesses is

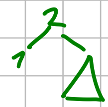
$$O\left(n \log n + m + \sum_i \log(1+z_i)\right).$$

③ Static Optimality

Seq. of accesses x_1 — $x_m \in X$

↳ frequences: $f: X \to \mathbb{N}$

↳ what if non-uniform?



T is some BST on X

$c_T(x) :=$ cost of accessing x in T

$=$ length of root-to-x path in T

total cost $= \sum_i c_T(x_i) = \sum_{x \in X} f(x) \cdot c_T(x)$

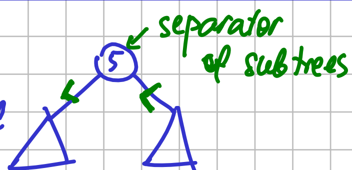
↳ want to minimize this cost

↳ statically optimal tree (can be found in $O(n^2)$ time using dyn. prog.)

Theorem: If $\forall x\ f(x) > 0$, then cost of x_1 — x_m in a Splay tree is O(cost in a static tree T) for arbitrary T.

④ Dynamic Optimality Conjecture

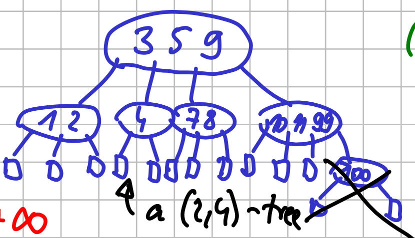
Multi-way Search Trees

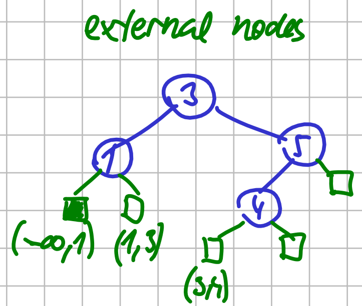


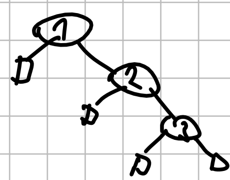
Df: MWST is a tree with external nodes where each node contains keys $x_1 < x_2 < \ldots < x_k$ and subtrees S_0 —— S_k such that $\forall i$

$x_i <$ all keys in $S_i < x_{i+1}$

$x_0 := -\infty$ $\qquad$ $x_{k+1} := +\infty$







Df: (a,b)-tree with parameters $a \geq 2$, $b \geq 2a-1$ is a MWST such that:

① every internal node has between a and b children except root has between 2 and b.

② all external nodes are on the same level.

Examples: (2,3) and (2,4)

Lemma: Height of an (a,b)-tree with n nodes is between $\log_b(n+1)$ and $\log_a\left(\frac{n+1}{2}\right)+1$.

Proof: Exercise.

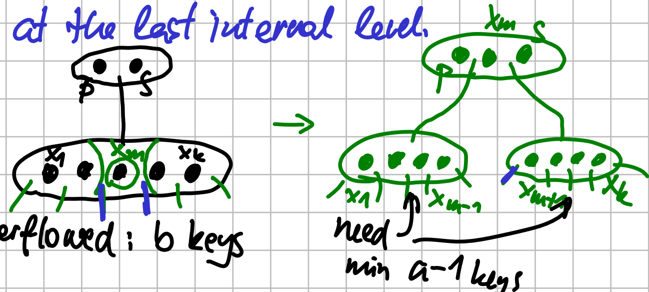
↳ so height is $\Omega(\log_b n)$ and $O(\log_a n)$

$\frac{\log n}{\log b}$ $\qquad$ $\frac{\log n}{\log a}$

Operations

Find walks a path from the root — $O\left(\frac{\log n}{\log a}\right)$ steps
in every step bin. search — $O(\log b)$
$\Bigg\}\ O\left(\log n \cdot \frac{\log b}{\log a}\right)$

Insert performs Find
adds a new key at the last internal level.
Fixing overflows by splitting nodes



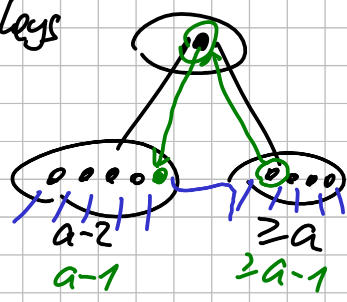
→ cascade splitting
↓
can lead to split of root

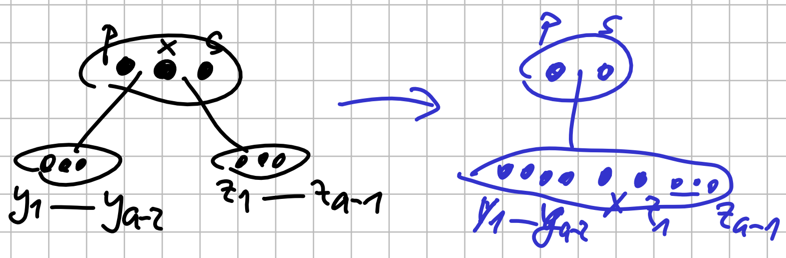
time: we work at $O\left(\frac{\log n}{\log a}\right)$ levels
at each level $O(b)$ work.
↳ total $O\left(\log n \cdot \frac{b}{\log a}\right)$

in total we need at least $(a-1) + 1 + (a-1)$
$= 2a-1$ keys
Ok thanks to $b \geq 2a-1$ by def.

Delete first we Find
& reduce to del. at the lowest level
(replace the key by its successor)

Handle underflow (node with $a-2$ keys)
— there is a sibling with at least a keys → steal one of his keys
— sibling with $a-1$ keys
↓
merge with the sibling





total
$(a-2) + 1 + (a-1)$ keys
‖
$2a-2$ keys
↓
$2a-1$ children
which is $\leq b$

... cascade, possibly up to the root...

time: at $O\left(\frac{\log n}{\log a}\right)$ levels
$O(b)$ per level
$\Bigg\}$ total $O\left(\log n \cdot \frac{b}{\log a}\right)$ ... as Insert

---

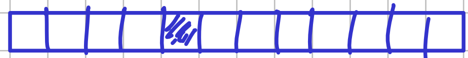
Choice of a, b

Find takes $O\left(\log n \cdot \frac{\log b}{\log a}\right)$
Ins, Del take $O\left(\log n \cdot \frac{b}{\log a}\right)$

both increasing in b → we want $b \in O(a)$
Find $O(\log n)$, Ins/Del $O\left(\log n \cdot \frac{a}{\log a}\right)$
grows with a
↳ we want to use as small a as possible

on the RAM choose (2,3) or (2,4)

Other cases: ① tree on the disk (disks are block-based, have slow seeks)



↳ set a, b such that a node fits in a block

for example: 4 KB blocks, 32-bit keys & pointers] ~8B per key

↳ $(256, 511)$-tree fits closely in 4KB

↳ in height 3 we reach at least $256^3 = 2^{24} \doteq 16M$ keys

↳ 2 accesses to disk per find

height 4 : $2^{32} \doteq 4G$

↳ 3 accesses per find

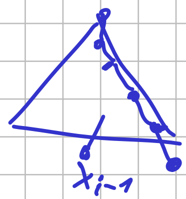
② cached memory

↳ we have typically 64Byte blocks

↳ e.g., $(4,7)$-trees.

Amortized Analysis of (a,b)-trees

Q: total cost of a sequence of m operations ?



we will measure just # changes } not counting time to find the location of the item

Next week: ① for m Inserts ... total cost is $O(m)$

② for m Ins/Dels ... $O(m)$ if $b \geq 2a$