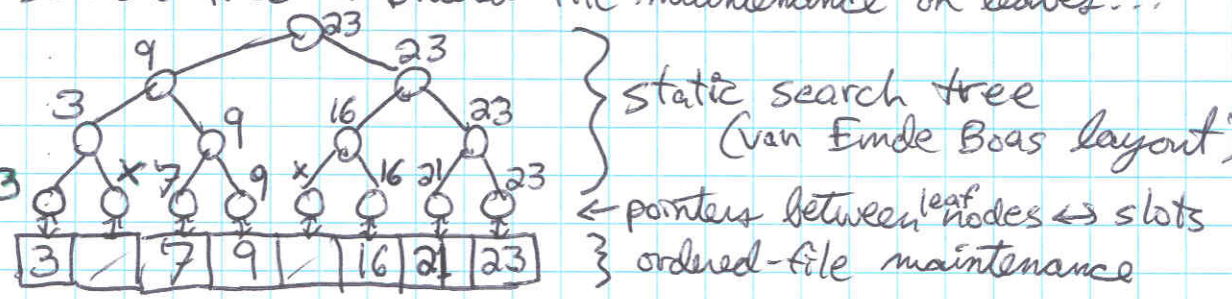
(Dynamic) B-trees

[Bender, Demaine, Farach-Colton 2000] | 6.897
[Bender, Duan, Iacono, Wu – SODA 2002] | L16.3
[Brodal, Fagerberg, Jacob – SODA 2002] | Apr. 23, 2003

- static search tree + ordered-file maintenance on leaves...



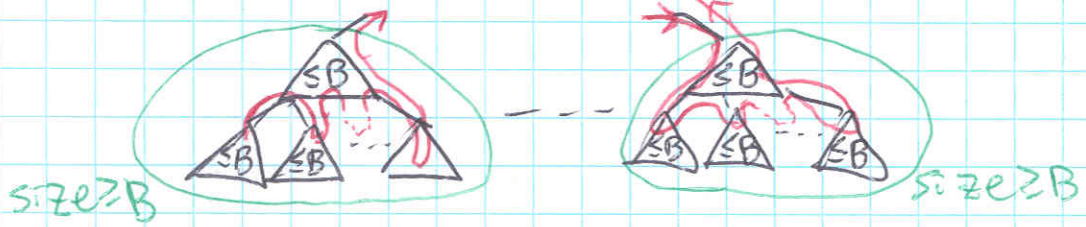
Invariant: each node stores max (descendent leaf keys)

- search can then use value @ right child to decide left/right.

- Insert:
  - either given appropriate slot or search for it
  - insert into ordered file
  - update corresponding leaves of changed cells & propagate labels up the tree in postorder traversal of changed leaves & their ancestors

Claim: $O(K/B + \log_B N)$ memory transfers for K changed leaves

Proof: as before, look at level of detail between $\le B$ & $> B$.
look at bottom 2 levels



if $N \ge 2B$, we use blocks optimally (update $\Theta(B)$ elements per block read)
$\Rightarrow O(\lceil K/B \rceil)$ in bottom 2 levels
remaining nodes in tree over updated leaves is $\le K/B$
$\Rightarrow O(1)$ memory transfers each is OK.!
also the path to root $\Rightarrow O(\log_B N)$ extra (as in static analysis) ∎

TOPS FORM 33041