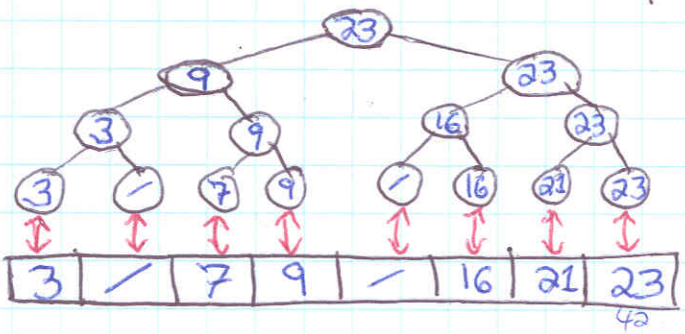
# Cache-oblivious B-trees — here follow [Bender, Duan, Iacono, Wu-SODA 2002]

6.897
L23.4
May 3, 2005

- Black box (for now): ordered file maintenance
  - store $N$ elements in specified order in $O(N)$ array (gaps)
  - insert element between two specified / delete element by moving elements in array interval of $O(\lg^2 N)$ am.
- build static search tree on top:



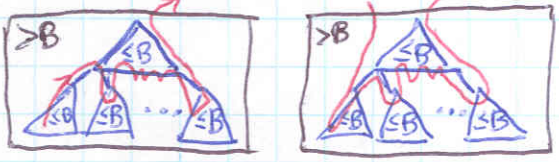

- static search tree (van Emde Boas layout)
- pointers connect leaf nodes & slots
- ordered file maintenance

- invariant: each node stores max {descendant leaf keys}
- search can decide direction by comparing right child's value
- insert(x):
  - search(x) finds predecessor or successor
  - insert x after/before pred/succ. in ordered file
  - update values in leaves corresponding to changed cells
  - propagate changes up tree, in post-order traversal of changed leaves & their ancestors
- delete(x) similar

- claim: if $K$ cells change then $O(K/B + \lg_B N)$ memory transfers

  proof: look at level of detail straddling $B$

  look at bottom 2 levels:




  within chunk of $>B$,
  jumping between $\leq 2$ subchunks of $\leq B$

  $\Rightarrow$ use $O(\text{chunk}/B)$ memory transfers within chunk (assuming $M \geq 2B$)

  $\Rightarrow O(K/B)$ memory transfers in bottom 2 levels

  - # nodes above these 2 levels $\leq \underbrace{K/B}_{\text{ancestors until lca can afford 1 transf. each}} + \underbrace{\lg N}_{\text{path to root } \lg_B N \text{ cost as before}}$

  $\Rightarrow O(K/B + \lg_B N)$ transfers