Delete / Insert

Once you find the correct index / location

Array: may need to shift $O(n)$ items

LL: can be done in $\Theta(1)$ time

Search:

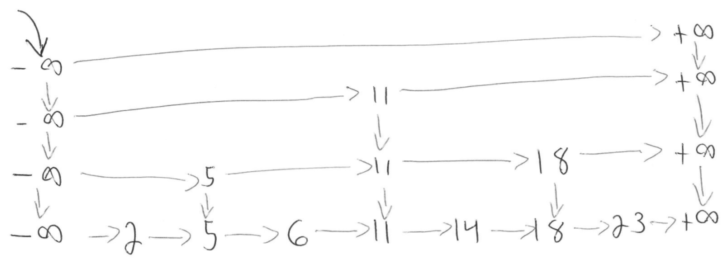
sorted array => binary search

LL => Skiplist!

Idea: need a faster way to traverse LL

- build a list with fewer nodes where some are skipped over
- then another, ...

Sentinels:
- markers to denote beginning & end
- helps simplify the code



Ideal layout.
- Difficult to maintain

Acts like binary search:
- is $k > 11$, search Right side (1 comparison)
- is $k < 11$, stay on Left half "

Tempted to doubly-link ← not necessary

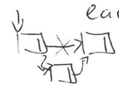
Search(87): stack P

S_0	83
S_1	83
S_2	65
S_3	$-\infty$

- getPredecessors
- P.top.after.key = k?

Stack contains largest key strictly less than k at each level

- need pointer to pred node for insert / delete



What is the probability of a tower of height 3?

at least $\frac{1}{8}$ or $\frac{1}{16}$ exactly

HHH...(H or T) HHHT

"exactly" or "at least"?

Expected Height of Skiplist

S_0: has all n keys

Expect S_1 to have $n/2$ keys

S_2 $n/4$

S_3 $n/8$

S_k $n/2^k$

+1 for sentinel only level

=> Expected height $O(\log n)$

Total # nodes $\leq \sum_{i=0}^{\infty} \frac{n}{2^i} = n \sum_{i=0}^{\infty} \frac{1}{2^i} = 2n$ + sentinals

- sum over levels

or $\sum_{i=0}^{k} \frac{n}{2^i} = 2n - \frac{n}{2^k}$

Runtime of operations:

Expected $O(n + \log n)$?

- Don't usually scan all n elements

Analysis

Drop-downs • always get to bottom level

=> height of skiplist

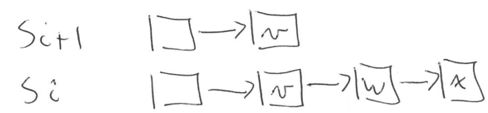
=> expected height $O(\log n)$

Step-forwards: expected # of forward steps on level S_i is at most 1

If $i = h$ (top level) => no step-forwards

Assume • $i < h$ and dropped down from level S_{i+1} at v

- w is node after v on S_i



What is the probability we step-forward from v to w?

Step forward to w on S_i means

- w did not exist on S_{i+1} ← or would have step-forward on S_{i+1}
- tower for w has height exactly i
  - aside: always enter a tower at the top
- this has probability $\frac{1}{2}$ P(tower w exactly height i | given has height at least i)
- prob to extend from S_i to S_{i+1} coin flip

$\Rightarrow$ step-forward with prob at most $\frac{1}{2}$

- even if w did ~~not~~ exist on S_{i+1} we may still drop-down (if $K \le w.key$)

Repeating this argument: prob step to x?

$(\frac{1}{2}$ to $w)(\frac{1}{2}$ to $x) = \frac{1}{4}$

and so on

$$\Rightarrow E[\#\text{ step-forwards}] = \sum_{\ell \ge 1} \frac{1}{2^\ell} \le 1$$

Self-Organizing Search

- messy-desk ≥ commonly required items on top of mess
- 80% of searches are on 20% of items

Exchange Proof

Current $\frac{2}{26} \cdot 1 + \frac{8}{26} \cdot 2 \ \underline{?}$ (A, B)

Swap(B, A) $\frac{8}{26} \cdot 1 + \frac{2}{26} \cdot 2$

MTF
- quick to steady state
- affected by rare lookup
  - lots of work to shuffle back
- Can show $C_{MTF} \le 2 \cdot C_{OPT}$

Transpose
- slow to steady state
- unaffected by rare lookup