Announcements

• Skip Lists
Skip Lists

• Problem: searching a linked list takes too long O(n)
• Want to search more quickly
• Solution: add more edges that let us skip through items
Skip List

S₀: -inf 2 5 7 9 +inf
S₁: -inf 5 9 +inf
S₂: -inf 5 +inf
Skip List

- Consists of a series of lists \( \{S_0, S_1, \ldots, S_h\} \)
- \( S_0 \) contains every item
- For \( i=1,\ldots,h-1 \) \( S_i \) contains a randomly selected subset of \( S_{i-1} \) plus \(-\infty\) and \(+\infty\)
- \( S_h \) contains only \(-\infty\) and \(+\infty\)
Positions

after(p) - position after p on the same level
before(p) - position before p on the same level
below(p) - position below p on the same tower
above(p) - position above p on the same tower
Searching for k in a Skip List

Finds the largest key <=k

Start at top-most level in the left most position p
while below(p) != null do
    p <- below(p)  // drop down
    while (key(after(p)) <= k do
        p <- after(p)  // scan forward
return p
Skip List

Search for 7

S_0: -inf → 2 → 5 → 7 → 9 → +inf
S_1: -inf → 5 → 9 → +inf
S_2: -inf → 5 → +inf
Skip List

Search for 7

$S_2$: -inf, 5, +inf

$S_1$: -inf, 5, 9, +inf

$S_0$: -inf, 2, 5, 7, 9, +inf
Skip List

Search for 7
Skip List

Search for 7
Skip List

Search for 7

S₀  -inf  2  5  7  9  +inf
S₁  -inf  5  9  +inf
S₂  -inf  5  +inf
Skip List

Search for 7

S₀ -inf 2 5 7 9 +inf
S₁ -inf 5 9 +inf
S₂ -inf 5 +inf
Insertion of $k$

$p =$ Search for $k$ using search procedure
Add $k$ after item $p$ at bottom level

while random() < 1/2 do
  while above($p$) == null do
    $p$ <- before($p$)
  $p$ <- above($p$)
insert after item $p$ at next higher level
Removal of k

1. Find k
2. Remove k from bottom level
3. Look at next level up
4. If k is present in this level, remove k from this level otherwise exit
5. Goto 3
Cost

• Expected height
Each level has half the expect number of entries as the previous one
\[ P_i \leq n/2^i \]

\[ \Rightarrow \text{Expected number of levels is } O(\log(N)) \]

Book has more formal reasoning
Search Time

- Outer loop executes $O(h)$ which with high likelihood is $O(\log n)$
- Likely to make $O(1)$ operations on given level
- Only considering keys on level $i$ between the current key and the next greater one on level $i+1$
- Half of these keys in the range $[k, \text{next}(k \text{ on level } i+1)]$ should appear in level $i+1$ and only $k$ and $\text{next}(k \text{ on level } i+1)$
- Expect to scan small constant number of keys on each level $= O(1)$
- Total search $O(\log n)$
Space Usage

• Bottom level $n$
• Next level $n/2$
• Next level $n/2^2$
• Sum of $n + n/2 + n/4... = n(1 + 1/2 + 1/4...) = 2n$
• Space = $O(n)$