

Ordered Collection Overview

Note Title

11/2/2007

Skip List Analysis

Analysis

(What is expected height (# of levels in tallest tower)?

X (How many times per level do we expect to move (or peak) forward)?

Last time argued that expect $n \cdot p^i$ elements in level i list.

Find i for which $n \cdot p^i = \frac{1}{p}$

$$n = \left(\frac{1}{p}\right)^{i+1}$$

$$\log_{1/p} n = i + 1$$

$$i = \log_{1/p} n - 1$$

roughly
 $\log_{1/p} n$
levels

For $p=1/4$,
level
when
you
only
expect 4
elements
to
remain

So at level $\log_{1/p} n - 1$ expect

only $1/p$ elements to remain

levels until this point $\log_{1/p} n$

0, 1, 2, ..., $\log_{1/p} n - 1$

Can prove

$$E[\text{height of tallest tower}] = \log_{1/p} n + \frac{1}{1-p}$$

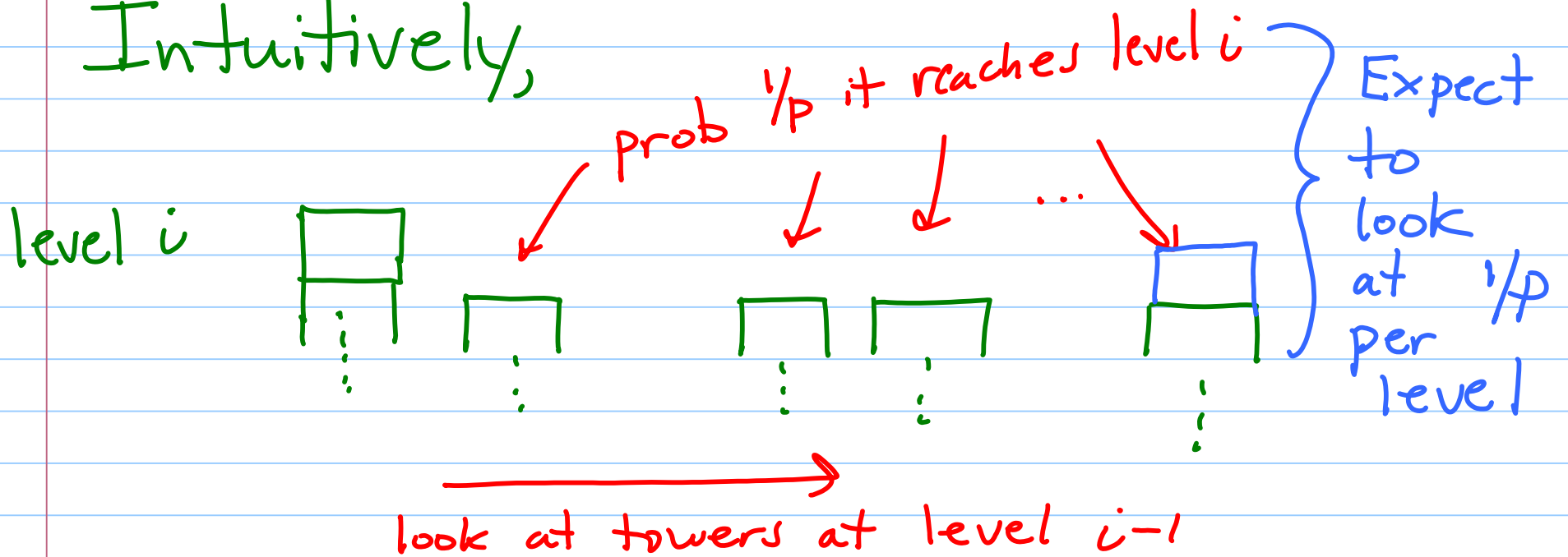
$$p = 1/4$$

$$\log_4 n + \frac{4}{3}$$

Search Cost

Page 615 goes over more formally.

Intuitively,



So expect $\log_{1/p} n + \frac{1}{1-p}$ levels $\leftarrow p = 1/4, \log_4 n + \frac{4}{3}$

+ expect to consider $1/p$ elements per level

$$\log_4 n = \frac{\log_2 n}{\log_2 4} = \frac{\log_2 n}{2}$$

$$\Rightarrow E[\text{search time}] \approx \frac{1}{p} \log_{1/p} n = \frac{1/p \log_2 n}{\log_2 1/p}$$

$$\text{For } p = 1/4, E[\text{search time}] \approx 2 \log_2 n$$

Space Usage

What is expected height of a tower

$$E[h] = \sum_{i=0}^{\infty} i p^{i-1} (1-p) = \frac{1}{1-p}$$

$i-1$ times
continue
(occurs with
prob p)

last time
stop
(occurs prob
 $1-p$)

next/prev refs (dominates space usage)

$$\approx n \cdot \frac{1}{1-p} \cdot 2 + 4 \left(\log_{1/p} n + \frac{1}{1-p} \right)$$

towers

expected height

next + prev per level of each tower

for head + tail

$$= \frac{2n}{1-p} + 4 \log_{1/p} n + \frac{4}{1-p}$$

next/prev refs in skiplist for $p=1/4$

$$2\frac{2}{3}n + 2\log_2 n + \frac{8}{3} \approx 2\frac{2}{3}n$$

left, right, parent refs in red-black tree

$$3n$$

parent, child, next, prev refs in B^+ -tree with $t=2$

On average $\sim n/2$ leaves + $\sim \frac{3n}{4}$ internal nodes


$$\Rightarrow \sim 3(n/2) + 5(3n/4) = \frac{21n}{4} = 5\frac{1}{4}n$$

parent + 4 child refs

Skip List overview

	As a function of p	$p = 1/2$	$p = 1/e$	$p = 1/4$
space usage	$2n/(1-p) + 4 \log_{1/p} n$	$4n + 4 \log_2 n$	$\approx 3.2n + 2.8 \log_2 n$	$\approx 2.67n + 2 \log_2 n$
search cost	$(\log_{1/p} n)/p$	$2 \log_2 n$	$\approx 1.88 \log_2 n$	$2 \log_2 n$

value of p to
minimize search
cost



Overview of Ordered Collection Data Structures

Key

- ⊙ Excellent
- Very Good
- Fair
- Method does nothing

Ordered
Collection
Methods

Method	SortedArray	BinarySearchTree	RedBlackTree	SplayTree	Btree	B+Tree	SkipList
add(o)	●	●	○	○	○	○	○
addAll(c), per element	○	●	○	○	○	○	○
clear(), per element	⊙	⊙	⊙	⊙	⊙	⊙	⊙
contains(o)	○	●	○	○	○	○	○
ensureCapacity(x)	●	•	•	•	•	•	•
getLocator(o)	○	●	○	○	○	○	○
iterator(), iteratorAtEnd()	⊙	○	○	○	○	⊙	⊙
max(),min()	⊙	○	○	○	○	⊙	⊙
prededessor(o)	⊙	○	○	○	○	⊙	⊙
remove(o), retainAll(c), per element	●	●	○	○	○	○	○
successor(o)	⊙	○	○	○	○	⊙	⊙

Key

- ⊙ Excellent
- Very Good
- Fair
- Method does nothing

Other Issues

Locator Methods

Method	SortedArray	BinarySearchTree	RedBlackTree	SplayTree	Btree	B+Tree	SkipList
typical space	⊙	○	○	○	○	●	○
randomized							✓
amortized approach (occasionally restructures)				✓			
fast access to recently accessed elements				✓			
designed to minimize disk pages read					✓	✓	
advance()	⊙	○	○	○	○	⊙	⊙
get()	⊙	⊙	⊙	⊙	⊙	⊙	⊙
hasNext(), next()	⊙	○	○	○	○	⊙	⊙
remove()	●	●	○	○	○	○	⊙
retreat()	⊙	○	○	○	○	⊙	⊙

