**Skip Lists**

The subway data structure

Searching in a linked list

$\Theta(n)$ worst case time

Even if it's sorted

Think of the list as a subway line.

(search item = desired stop)
Search: start at beginning of express line, travel until finding desired stop or passing it if not found, back up one stop and use the slow line.

How many stops should L2 have?
How should we distribute them?
Uniform distribution

arbitrary gap choice

MAX
Search time: \(|L_2| + \frac{|L_1|}{|L_2|} + O(1)\)

\(|L_1| = n\); minimize \(|L_2| + \frac{n}{|L_2|}\) \implies \(|L_2| = \sqrt{n}\) \implies \text{cost} = 2\sqrt{n} + O(1)

What if we had 3 lines? Add an ultra-express.

Could get cost \(\leq \frac{n}{L_2} + \frac{L_2}{L_3} + L_3 + O(1)\) \rightarrow \text{minimized at} \: \frac{n}{n^{2/3}} + \frac{n^{2/3}}{n^{1/3}} + n^{1/3} = 3n^{1/3}
How many lists should we use?

If we would like $O(\log n)$ search time, use at most $k = O(\log n)$ lists.

$$n^{\frac{1}{k}} = 2^{\log n^{\frac{1}{k}}} = 2^{\frac{1}{k} \log n} = 2$$ if $k = \log n$  
$$< 2$$ if $k > \log n$

$$k \cdot n^{\frac{1}{k}} \sim 2 \cdot \log n$$ if $k = o(\log n)$

Any 2 consecutive level sizes have constant ratio, $n^{\frac{1}{k}}$

For $k = \log n$ levels, ratio $= 2$

\[ \text{size of structure} \approx 2n \]
notice that every level has a node at extreme left.
DYNAMIC SKIP LISTS

Assume we have a skip list with \( \log n \) levels. We want to insert a new element, \( x \).

1. Find the position of \( x \): search.
2. Insert \( x \) in \( L_1 \).

\[ \exists \text{walk} \leq 2 \text{ nodes per level} \]

Correcting levels can take \( O(n) \) time if we are rigid about structure.
One fix: allow the ratio between levels to vary (within 2~4).

4 won't affect the search time, using big-O.

4, when we have 3 items between nodes in level above, add a pointer to level above & create a node there.
Insert. Do nothing else.
Randomized insertion of $x$

First insert in $L_1$

Then flip a coin: If $H$, promote $x$ to next level, and repeat.
H vs T
(else do nothing)
If we reach the top and want to promote, make new level

Clearly there are 2 main problems.
1) Too many consecutive $H$ : builds dense vertical links.
2) Too many $T$ : generates a linked list
Insert 44, flip T

make sure every row has a starting “station”

-∞               44
Insert 9, flip H
Insert 9, flip H, promote 9, flip T

-∞ 9

-∞ 9 44
Insert 26, T

\[ \infty \quad 9 \]

\[ \infty \quad 9 \quad 26 \quad 44 \]
$50 \quad H \rightarrow T$

- $-\infty \quad 9 \quad 26 \quad 44 \quad 50$
$12 \quad H \rightarrow T$

$-\infty \quad 9 \quad 12 \quad \rightarrow \quad 50$

$-\infty \quad 9 \quad 12 \quad 26 \quad 44 \quad 50$
<table>
<thead>
<tr>
<th>-∞</th>
<th>9</th>
<th>12</th>
<th>26</th>
<th>37</th>
<th>44</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
52 T

-∞ 9 12 26 37 44 S0 S1 S2
Deletion: find element, delete in all levels
Claim: \( \text{every search costs } O(\log n) \) with high probability

expected time complexity, but with guarantees.

...as opposed to quicksort which has expected \( \Theta(n \log n) \)

but in \( n \) quicksorts, \( \Theta(\log n) \) of them could take \( O(n^2) \) time

without violating the bound.

average time = \( \frac{1}{n} \cdot \left[ \Theta(\log n) \cdot n^2 + (n - \Theta(\log n)) \cdot n \log n \right] \)

High probability for event \( E \):

for any \( \alpha > 1 \), \( \exists \) constants s.t. \( \text{prob}(E) \geq 1 - O \left( \frac{1}{n^{\alpha}} \right) \)
High probability for event $E$:

For any $\alpha > 1$, there exist constants such that $\text{prob}(E) \geq 1 - O\left(\frac{1}{n^\alpha}\right)$

$$\text{prob}(\overline{E}) = O\left(\frac{1}{n^\alpha}\right)$$

Example: $\text{pr}\{E\} = \frac{1}{2}$ is not high even if it is tempting to say $0.5$ is a constant and so is $\frac{1}{n^\alpha}$

$$\text{pr}\{E\} = \frac{1}{2} = 1 - \left(\frac{1}{2}\right) \quad \text{We are not saying} \quad 2 = O(n^\alpha)$$

$$\text{pr}\{E\} = 1 - \frac{1}{x} \quad \text{is high if} \quad x = \Omega\left(n^\alpha\right) \quad \text{... ex.} \quad \text{pr}\{E\} = \frac{1}{2^n}$$
High probability for event $E$:

for any $\alpha > 1$, $\exists$ constants s.t. $\text{prob}(E) \geq 1 - O\left(\frac{1}{n^{\alpha}}\right)$

$\sim$ for any $\alpha$, $\text{prob}(\tilde{E}) \leq \frac{1}{n^{\alpha}}$

We will look at event $X$: every search costs $O(\log n)$

$\bar{X}$: some search costs $\omega(\log n)$

Intuition: $\bar{X}$ happens if we flip $H$ too often when inserting or if we insert many elements and get $T$. 

Unlikely
$\bar{X}$: some search costs $w(\log n)$ depends on construction & on # searches
we want $\text{prob}(\bar{X}) \leq \frac{1}{n^a}$

One of two ways to get $\bar{X}$: flip H too often when inserting some $e_i$;
$\Rightarrow$ obtain $w(\log n)$ levels in skiplist.

We know that $\Pr[e_i \text{ generates } w(\log n) \text{ levels}] = \bar{X}_i$ is low

$\bar{X}_i \leq \frac{1}{2^{w(\log n)}}$ to flip H $w(\log n)$ times
$2^{w(\log n)} = w(2^{\log n}) = w(n)$

The question is: how many elements can we insert and still have $O(\log n)$ levels WHP?
Proof

Boole's inequality \( [\text{appendix C, CLRS}] \)

\[
\Pr\{E_1 \cup E_2 \cup \ldots \cup E_k\} \leq \Pr\{E_1\} + \Pr\{E_2\} + \ldots + \Pr\{E_k\}
\]

does not assume independence

\[\Pr\{\overline{X}_i \cup \ldots \cup \overline{X}_k\} = \Pr\{\text{some insert takes long}\}\]

\(\Delta \) we will use \(E_i = \overline{X}_i\)
Boole's inequality \[\text{[appendix C, CLRS]}\]

\[\Pr\{E_1 \cup E_2 \cup \cdots \cup E_k\} \leq \Pr\{E_1\} + \Pr\{E_2\} + \cdots + \Pr\{E_k\}\]

does not assume independence

\[\overline{E_1} \cap \overline{E_2} \cap \cdots \cap \overline{E_k} = \text{probability of no } E_i \text{ occurring}\]

\[\overline{X_i}\]

we want this
**Proof**

Boole’s inequality \([\text{appendix C, CLRS}]\)

\[
\Pr\{E_1 \cup E_2 \cup \cdots \cup E_k\} \leq \Pr\{E_1\} + \Pr\{E_2\} + \cdots + \Pr\{E_k\}
\]

*does not assume independence*

\[
\overline{E_1} \cap \overline{E_2} \cap \cdots \cap \overline{E_k} = \text{probability of no } E_i \text{ occurring}
\]

\[
= 1 - \Pr\{E_1 \cup E_2 \cup \cdots \cup E_k\}
\]

\[
\geq 1 - \Pr\{E_1\} + \Pr\{E_2\} + \cdots + \Pr\{E_k\}
\]
Proof

Boole's inequality \[ \text{(appendix C, CLRS)} \]

\[ \Pr \{E_1 \cup E_2 \cup \cdots \cup E_k\} \leq \Pr \{E_1\} + \Pr \{E_2\} + \cdots + \Pr \{E_k\} \]

does not assume independence

Suppose we have a polynomial number of events: \( k = n^c \), \( c = o(1) \)

each \( E_i \) occurs \( \text{WHP} \)

\[ \Pr \{\overline{X_i}\} = O(\frac{1}{n^c}) = \Pr \{E_i\} \]

\[ \overline{E_1} \cap \overline{E_2} \cap \cdots \cap \overline{E_k} = \text{probability of no } E_i \text{ occurring} \]

\[ = 1 - \Pr \{E_1 \cup E_2 \cup \cdots \cup E_k\} \]

\[ \geq 1 - \Pr \{E_1\} + \Pr \{E_2\} + \cdots + \Pr \{E_k\} \]

\[ = 1 - n^c \cdot O\left(\frac{1}{n^c}\right) = 1 - O\left(\frac{1}{n^{c-\epsilon}}\right) \text{ WHP} \]
Proof

Boole’s inequality \([\text{appendix C, CLRS}]\)

\[
\Pr\{\overline{X}_1 \cup \overline{X}_2 \cup \ldots \cup \overline{X}_k\} \leq \Pr\{\overline{X}_1\} + \Pr\{\overline{X}_2\} + \ldots + \Pr\{\overline{X}_k\}
\]

does not assume independence

Suppose we have a polynomial number of events: \(k = n^c, c = o(1)\)

\(X_1 \land X_2 \land \ldots \land X_k\) = probability of no \(\overline{X}_i\) occurring

\[
= 1 - \Pr\{\overline{X}_1 \cup \overline{X}_2 \cup \ldots \cup \overline{X}_k\}
\]

\[
\geq 1 - \Pr\{\overline{X}_1\} + \Pr\{\overline{X}_2\} + \ldots + \Pr\{\overline{X}_k\}
\]

\[
= 1 - n^c \cdot O\left(\frac{1}{n^c}\right) = 1 - O\left(\frac{1}{n^c - c}\right) \text{ WHP}
\]

each \(X_i\) occurs WHP

\[
\Pr\{\overline{X}_i\} = O\left(\frac{1}{n^c}\right)
\]
Suppose we have a polynomial number of events: $k = n^c$, $c = o(1)$

each $X_i$ occurs $\text{WHP}$

$$\Pr\{\neg X_i\} = O\left(\frac{1}{n^\alpha}\right)$$

$$X_1 \land X_2 \land \ldots \land X_k = \text{probability of no } \neg X_i \text{ occurring}$$

$$= 1 - \Pr\{\neg X_1 \lor \neg X_2 \lor \ldots \lor \neg X_k\}$$

$$\geq 1 - \Pr\{\neg X_1\} + \Pr\{\neg X_2\} + \ldots + \Pr\{\neg X_k\}$$

$$= 1 - n^c \cdot O\left(\frac{1}{n^\alpha}\right) = 1 - O\left(\frac{1}{n^{\alpha - c}}\right) \text{WHP}$$

We just have to make sure we show that any $\Pr\{\neg X_i\} = O\left(\frac{1}{n^\alpha}\right)$ for $\alpha > c + 1$

For $n^c$ inserts we will know WHP that # levels $= O(\log n)$
$\alpha$ is not a parameter to choose, that will affect the algorithm. It is just a measurement of the probability of obtaining higher time comp.

example: $\Pr\{\text{search takes } > 1000 \log n\} \sim 1 - \frac{1}{n^{1000}}$
Claim: WHP, \# levels = O(\log n) \leq c \cdot \log n

Proof: \Pr\{\text{claim is wrong}\} = \Pr\{\# \text{levels} > c \cdot \log n\}
\leq n \cdot \Pr\{X \text{ is promoted } > c \cdot \log n \text{ times}\} \quad \text{by Boole}
\leq n \cdot \left(\frac{1}{2}\right)^{c \log n} \quad \text{flip H \ clogn times in a row}
= n \cdot \frac{1}{n^c} = \frac{1}{n^{c-1}} \\
\log_a c \log n = c \log n \cdot \log a = \log n c \log a

Let \alpha = c - 1. \ \text{QED}

However, search also depends on the (horizontal) buildup of large linked lists.
\[(\frac{1}{2})^\text{c logn}\] : chance of reaching height c logn

What if we are getting Too low a height? (at too many nodes)

Don't care how these got promoted here. Once we look at top level, each has 50-50 chance of being promoted. Unlikely to see many nodes here

top level →

expect the right #nodes/level. But could still get gaps →
Search path \( \Rightarrow \) to find \( z \) \( \sim \)

If \( p \) is a bend, we know that it was not promoted.
Otherwise we would have searched further \( \Rightarrow \) in the level above.

The same holds for all nodes between \( p \) and \( q \).

If we walk left on the path from \( z \) until we find a bend, we follow a horizontal segment of unpromoted nodes.
Each one had a so-so chance of being promoted. \( \Rightarrow \) segment length \( = O(\log n) \)

WHP
Path from \( z \) to "start"

\[ \text{# of UP moves} = \text{# levels} \leq c \cdot \log n \text{ WHP} \]

\[ z \leq \text{# total moves} \leq \text{# coin flips until getting } c \cdot \log n \text{ "H".} \]

because of previous analysis.

\( H \) at high height implies many other consecutive \( H \) below

obtaining \( > c \log n \) \( H \) is very unlikely
Path from $z$ to "start"

$\#$ of UP moves = $\#$ levels $\leq c \cdot \log n$ w.h.p

$\exists \#$ total moves $\leq \#$ coin flips until getting $c \cdot \log n$ "H".

Consider the odds of $\#$ total moves $> 10 \cdot c \log n \Rightarrow$ flip $10 \cdot c \log n$ coins

$\Pr$ exactly $\frac{c \log n}{c \log n} = \binom{10 \log n}{\log n} \cdot \left(\frac{1}{2}\right)^{c \log n} \cdot \left(\frac{1}{2}\right)^{9 \log n} \Rightarrow \Pr.$ the rest are T.

$\Pr$. obtaining $H$ $c \log n$ times.
Path from \( z \) to "start"

\[
\text{# of UP moves} = \text{# levels} \leq c \cdot \log n \text{ w.h.p}
\]

\[
\exists \ # \text{total moves} \leq \# \text{coin flips until getting c \cdot \log n "H".}
\]

Consider the odds of \#total moves > 10 \cdot c \cdot \log n = \text{flip 10 \cdot c \cdot \log n coins \ w/} \leq c \cdot \log n "H".

\[
\Pr \frac{z}{c \cdot \log n} \geq \frac{c \cdot \log n}{10} (\text{c \cdot \log n}) \cdot \left( \frac{1}{2} \right)^{c \cdot \log n} \cdot \left( \frac{1}{2} \right)^{9 \cdot c \cdot \log n}
\]

\[
\Pr \text{the rest are T.}
\]

ways to rearrange c \cdot \log n "H" in all flips positions

\[
\Pr \text{obtaining X c \cdot \log n times, anything}
\]
Path from $z$ to "start"

\# of UP moves = \# levels $\leq c \cdot \log n$ w.h.p

$z \implies \# \text{total moves} \leq \# \text{coin flips until getting } c \cdot \log n \text{ "H".}$

Flip $10 \cdot c \log n$ coins.

$\Pr \mathcal{E} \leq c \log n H^3 \leq \binom{10 c \log n}{c \log n} \cdot (\frac{1}{2})^{9 c \log n}$

$\left(\frac{y}{x}\right)^x \leq \left(\frac{e}{x}\right)^x$

$(9 - \log 10e) \cdot c = \alpha$

\[
\frac{1}{2^{(k-1-\log ke)c\log n}} \sim \frac{1}{n^{\alpha}}
\]

| $\frac{1}{n^{\alpha}}$ choose $\frac{1}{n^{\alpha}}$: \Pr fail $\implies$ search time to expect |