SKIP LISTS
Skip Lists

Searching in a linked list
\( \Theta(n) \) worst case time
even if it's sorted
Skip Lists

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)

Manhattan
C, E lines

14  23  34  42  50  59  66  72  79
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
Search:
- start at beginning of express line
- travel until finding desired stop or passing it
Search: start at beginning of express line
travel until finding desired stop or passing it
if not found, back up one 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

$L_2$

$L_1$

arbitrary gap choice
Uniform distribution

$$L_2$$

$$L_1$$

Maximum search time: \[ |L_2| + \frac{|L_1|}{|L_2|} + O(1) \]
Uniform distribution

\[ |L_2| + \frac{|L_1|}{|L_2|} + O(1) \]

**Max Search Time:**

\[ |L_1| = n \quad \text{minimize} \quad |L_2| + \frac{n}{|L_2|} \]
Uniform distribution

\[ \text{arbitrary gap choice} \]

**Max Search Time:** \[ |L_2| + \frac{|L_1|}{|L_2|} + O(1) \]

\[ |L_1| = n \] minimize \[ |L_2| + \frac{n}{|L_2|} \] \[ \Rightarrow |L_2| = \sqrt{n} \] \[ \Rightarrow \text{cost} = 2\sqrt{n} + O(1) \]
Uniform distribution

arbitrary gap choice

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

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

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

$L_3$
$L_2$
$L_1$
Uniform distribution

\[ \text{arbitrary gap choice} \]

\[ |L_2| + \frac{|L_1|}{|L_2|} + O(1) \]

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

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

Could get cost \( \leq \frac{n}{L_2} + \frac{L_2}{L_3} + L_3 + O(1) \)
Uniform distribution

\[ L_2 \xrightarrow{\text{arbitrary gap choice}} \]
\[ L_1 \]

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

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

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

\[ \text{Could get cost } \leq \frac{n}{L_2} + \frac{L_2}{L_3} + L_3 + O(n) \Rightarrow \text{minimized at } \frac{n}{n^{2/3}} + \frac{n^{2/3}}{n^{1/3}} + n^{1/3} = 3n^{1/3} \]
k lists: \( O(k \cdot n^{\frac{1}{k}}) \) search

How many lists should we use?
How many lists should we use?

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

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

\[
n^{\frac{1}{k}} = 2^{\frac{\log n}{k}} = 2^{\frac{1}{k} \log n}
\]
How many lists should we use?

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

$$n^{1/k} = 2^{\log n^{1/k}} = 2^{\frac{1}{k} \log n} = 2 \text{ if } k = \log n < 2 \text{ if } k > \log n$$
How many lists should we use?

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

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

$$k \cdot n^{1/k} \sim 2 \cdot \log n \quad \text{if} \quad k \approx \log n$$
How many lists should we use?

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

$$n^{1/k} = 2^{\log_2 n^{1/k}} = 2^{\frac{1}{k} \log_2 n} = 2 \text{ if } k = \log n < 2 \text{ if } k > \log n$$

$$k \cdot n^{1/k} \sim 2 \cdot \log n \text{ if } k = \log n$$

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

For $k = \log n$ levels, ratio = 2.
How many lists should we use?

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

$\sqrt[k]{n} = 2^{\log n^{1/k}} = 2^{\frac{1}{k} \log n} = 2$ if $k = \log n < 2$ if $k > \log n$

$k \cdot n^{1/k} \sim 2 \cdot \log n$ if $k = \log n$

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

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

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

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

1. Find position of $x$: search $L_1$.
2. Insert $x$ in $L_1$.  

$\text{time}$?
Assume we have a skip list with log n levels. We want to insert a new element, x.

1. Find position of x: search
2. Insert x in L_1.

?walk ≤ 2 nodes per level
descend log n levels
DYNAMIC SKIP LISTS

Assume we have a skip list w/ log n levels.
We want to insert a new element, x.

1. find position of x: search
2. insert x in L₁.

?[walk ≤ 2 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)

won't affect the search time, using big-O.
One fix: allow the ratio between levels to vary (within 2~4).

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

- When we have 3 items between nodes in level above, add a pointer to level above & create a node there.
insert. Do nothing else
insert. Promote.
Randomized insertion of $x$

First insert in $L_1$

Then flip a coin: if H, promote $x$ to next level, and repeat.
Randomized insertion of $x$

First insert in $L_1$

Then flip a coin: If $H$, promote $x$ to next level, and repeat.

$H \text{ vs } T$

(else do nothing)

If we reach the top and want to promote, make new level
Randomized insertion of $x$

First insert in $L_1$

Then flip a coin:
- If $H$, promote $x$ to next level, and repeat.
- Else do nothing.

If we reach the top and want to promote, make new level.

Clearly there are 2 main problems.

1)
2)
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

2) Too many T

why?
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"
Insert 9, flip H
Insert 9, flip H, promote 9, flip T

-∞ 9

-∞ 9 44
$\inf \ 9$

$\inf \ 9 \ 26 \ 44$
12 H → T

-∞ 9 12 26 44 50
$37 \ T$

-∞  9  12  26  37  44  50
51 \text{T}

-\infty \quad 9 \quad 12 \quad \ldots \quad 50

-\infty \quad 9 \quad 12 \quad 26 \quad 37 \quad 44 \quad 50 \quad 51
<table>
<thead>
<tr>
<th>-∞</th>
<th>9</th>
<th>12</th>
<th>26</th>
<th>37</th>
<th>44</th>
<th>50</th>
<th>51</th>
<th>52</th>
</tr>
</thead>
</table>

52 T
Deletion: find element, delete in all levels
Claim: every search costs $O(\log n)$ with high probability expected time complexity, but with guarantees.
Claim: 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.
Claim: 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 = $\frac{1}{n} \cdot \left[ \Theta(\log n) \cdot n^2 + (n - \Theta(\log n)) \cdot n \log n \right]$
Claim: 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.

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

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

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

$\text{prob}(\overline{E}) = O\left(\frac{1}{n^\alpha}\right)$
High probability for event $E$:

For any $\alpha > 1$, exist constants s.t. $\Pr(E) \geq 1 - O\left(\frac{1}{n^\alpha}\right)$.

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

Example: $\Pr\{E\} = \frac{1}{2}$ is not high, even if it is tempting to say 0.5 is a constant and so is $\frac{1}{n^\alpha}$. 
High probability for event $E$:

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

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

Example: $p^c(E) = \frac{1}{2}$ is not high

Even if it is tempting to say 0.5 is a constant and so is $\frac{1}{n^{\alpha}}$

$p^c(E) = \frac{1}{2} = 1 - \left(\frac{1}{2}\right)$ We are not saying $2 = O(n^{\alpha})$
High probability for event $E$:

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

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

**Ex:** $\text{pr}\{E\} = \frac{1}{2}$ is not high

even if it is tempting to say 0.5 is a constant and so is $\frac{1}{n^\alpha}$

$$\text{pr}\{E\} = \frac{1}{2} = 1 - \left(\frac{1}{2}\right)$$

We are not saying $2 = O(n^\alpha)$

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

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

$\leadsto$ for any $\alpha$, $\text{prob}(\bar{E}) \leq \frac{1}{n^\alpha}$
High probability for event $E$:

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

\[
\sim \text{ for any } \alpha, \quad \Pr(\bar{E}) \leq \frac{1}{n^\alpha}
\]

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

$\bar{X}$: some search costs $\omega(\log n)$
High probability for event $E$:

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

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

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.
High probability for event $E$: 
for any $\alpha > 1$, there exist constants s.t. $\Pr(E) \geq 1 - O\left(\frac{1}{n^{\alpha}}\right)$
~ for any $\alpha$, $\Pr(\overline{E}) \leq \frac{1}{n^{\alpha}}$

We will look at event $X$: every search costs $O(\log n)$
$\overline{X}$: some search costs $\omega(\log n)$

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

Unlikely
$X: \text{ some search costs } w(\log n) = \text{ depends on } \ ? ?$
$X$: some search costs $w(\log n)$ depends on construction & on # searches

$X = \{X_1 \cup X_2 \cup \ldots \cup X_k\}$
\( \bar{X} \): some search costs \( w(\log n) \) depends on construction 

we want \( \text{prob}(\bar{X}) \leq \frac{1}{n^\alpha} \) 

\( \bar{X} = \{ \bar{X}_1, u \bar{X}_2 u \ldots u \bar{X}_K \} \)
$X$: some search costs $w(\log n)$: depends on construction & on #searches

we want $\Pr(X) \leq \frac{1}{n^\alpha}$

One of two ways to get $X$: flip $H$ too often when inserting some $e_i$

$\Rightarrow$ obtain $w(\log n)$ levels in skiplist
\( \bar{X} : \) some search costs \( w(\log n) \) depends on construction & on # searches
\[ \bar{X} = \bar{X}_1 \cup \bar{X}_2 \cup ... \cup \bar{X}_k \]

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

We know that \( \Pr[\exists e_i \text{ generates } w(\log n) \text{ levels}] = \bar{X}_i \) is low
\( \bar{X}_i \leq \frac{1}{2w(\log n)} \) to flip \( H \) \( w(\log n) \) times
\[ 2^{w(\log n)} = \omega(2^{\log n}) = \omega(n) \]
\( \bar{X} : \) 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 skip list

We know that \( \text{pr} \{ \exists e_i \text{ generates } w(\log n) \text{ levels} \} = \bar{X} ; \) is low

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

The question is: how many elements can we insert and still have \( O(\log n) \) levels WHP?
Boole's inequality [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
Boole's inequality \[ \Pr\{E_1 \cup E_2 \cup \cdots \cup E_k\} \leq \Pr\{E_1\} + \Pr\{E_2\} + \cdots + \Pr\{E_k\} \]
does not assume independence

Notice we care about
\[ \Pr\{\overline{X}_1 \cup \cdots \cup \overline{X}_k\} = \Pr\{\text{some insert takes long}\} \]

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

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

\(\text{does not assume independence}\)

\[
\overline{E}_1 \cap \overline{E}_2 \cap \cdots \cap \overline{E}_k = \text{probability of no } E_i \text{ occurring}
\]
Proof

Boole's inequality \[appendix C, CLRS\]

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

does not assume independence

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

\[\leftrightarrow \bar{x_i}\]

we want this
Boole’s inequality \cite[appendix C, CLRS]{Boole}

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

does not assume independence

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

\[
= 1 - \Pr\{E_1 \cup E_2 \cup \ldots \cup E_k\}
\]
Proof of Boole's inequality: 

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

does not assume independence

\[ \bar{E_1} \cap \bar{E_2} \cap \ldots \cap \bar{E_k} \] = probability of no \( E_i \) occurring

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

\[ \geq 1 - \Pr\{E_1\} + \Pr\{E_2\} + \ldots + \Pr\{E_k\} \]
Boole's inequality

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

does not assume independence

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

Each \( E_i \) occurs WHP \( \Pr \{ \overline{X}_i \} = O(\frac{1}{n^c}) = \Pr \{ E_i \} \)

\[ \overline{E}_1 \cap \overline{E}_2 \cap \ldots \cap \overline{E}_k = \text{probability of no } E_i \text{ occurring} \]

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

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

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

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

\[
\Pr \{\bar{X}_i\} = O\left(\frac{1}{n^c}\right) = \Pr \{\bar{E}_i\}
\]

\[
\bar{E}_1 \cap \bar{E}_2 \cap \ldots \cap \bar{E}_k = \text{probability of no } E_i \text{ occurring}
\]

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

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

\[
= 1 - n^c \cdot O\left(\frac{1}{n^c}\right)
\]
Boole's inequality  

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

does not assume independence

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

Each \( E_i \) occurs WHP

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

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

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

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

\[
= 1 - n^c \cdot O\left(\frac{1}{n^c}\right) = 1 - O\left(\frac{1}{n^{c-c}}\right) \text{ WHP}
\]
Boole's inequality \cite{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: \( k = n^c \), \( c = o(1) \)

Each \( X_i \) occurs \textbf{WHP} \( \Pr \{ \overline{X}_i \} = o\left( \frac{1}{n^c} \right) \)

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

\[
= 1 - \Pr \{ \overline{X}_1 \cup \overline{X}_2 \cup \ldots \cup \overline{X}_k \} \\
\geq 1 - \Pr \{ \overline{X}_1 \} + \Pr \{ \overline{X}_2 \} + \ldots + \Pr \{ \overline{X}_k \} \\
= 1 - n^c \cdot O\left( \frac{1}{n^c} \right) = 1 - o\left( \frac{1}{n^c} \right) \text{WHP}
\]
Suppose we have a polynomial number of events: \( k = n^c \), \( c = O(1) \)

Each \( X_i \) occurs WHP

\[
Pr\{\overline{X_i}\} = O\left(\frac{1}{n^\alpha}\right)
\]

\( X_1 \cap X_2 \cap \ldots \cap X_k \) = probability of no \( X_i \) occuring

\[
= 1 - Pr\{\overline{X_1} \cup \overline{X_2} \cup \ldots \cup \overline{X_k}\}
\]

\[
\geq 1 - Pr\{\overline{X_1}\} + Pr\{\overline{X_2}\} + \ldots + Pr\{\overline{X_k}\}
\]

\[
= 1 - n^c \cdot O\left(\frac{1}{n^\alpha}\right) = 1 - O\left(\frac{1}{n^{\alpha-c}}\right)\] WHP
Suppose we have a polynomial number of events: \( k = n^c \), \( c = o(1) \).

Each \( X_i \) occurs \textbf{WHP}.

\[
\Pr\{\bar{X}_i\} = O\left(\frac{1}{n^\alpha}\right)
\]

Let \( X_1 \land X_2 \land \cdots \land X_k \) be the probability of no \( \bar{X}_i \) occurring.

\[
X_1 \land X_2 \land \cdots \land X_k = 1 - \Pr\{\bar{X}_1 \lor \bar{X}_2 \lor \cdots \lor \bar{X}_k\}
\]

\[
\geq 1 - \Pr\{\bar{X}_1\} + \Pr\{\bar{X}_2\} + \cdots + \Pr\{\bar{X}_k\}
\]

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

We just have to make sure we show that any \( \Pr\{\bar{X}_i\} = O\left(\frac{1}{n^\alpha}\right) \) for \( \alpha \gg c + 1 \).

For \( n^c \) inserts we will know WHP that \( \#\text{ levels} = O(\log n) \).
$\alpha$ 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: $\Pr\{\text{search takes } > 1000 \cdot \log n\} \approx 1 - \frac{1}{n^{1000}}$
Claim: WHP, \#levels = O(log n) \leq c \cdot log n
Claim: WHP, \( \# \text{levels} = O(\log n) \leq c \cdot \log n \)

Proof: \(\Pr\{\text{claim is wrong}\} = \Pr\{\# \text{levels} > c \cdot \log n\}\)
\(\leq n \cdot \Pr\{x \text{ is promoted } > c \cdot \log n \text{ times}\} \quad \text{by Boole}\)

↑ for \(n\) inserts
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 (\frac{1}{2})^{c \log n} \quad \text{flip H \ log n times in a row}
Claim: WHP, \# levels = O(log n) ≤ 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 (\frac{1}{2})^{c \log n} \quad \text{flip H c \log n times in a row}

= n \cdot \frac{1}{n^c} = \frac{1}{n^{c-1}} \quad \log_a c \log n = c \log n \cdot \log_a n = \log_{n^c}
Claim: WHP, \# levels = \( \Theta(\log n) \) \( \leq c \cdot \log n \)

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 \cdot \log n} \quad \text{flip H \(c \cdot \log n\) times in a row} \)
\( = n \cdot \frac{1}{n^c} = \frac{1}{n^{c-1}} \quad \log a^{c \cdot \log n} = c \cdot \log n \cdot \log a = \log n^{c \cdot \log a} \)

Let \( \alpha = c - 1 \). QED

However, search also depends on the (horizontal) buildup of large linked lists.
\( \left( \frac{1}{2} \right)^{c \log n} \): chance of reaching height clog n
\( \left( \frac{1}{2} \right)^{c \cdot \log n} \): chance of reaching height \( c \cdot \log n \)

What if we are getting too low a height? (at too many nodes)
\[
\left(\frac{1}{2}\right)^{c \log n} \quad \text{chance of reaching height } c \log n
\]

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

top level \quad \text{low height}
\[
\left(\frac{1}{2}\right)^{c \cdot \log n} \quad \text{chance of reaching height } c \cdot \log n
\]

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.

\text{top level} \rightarrow \ldots \ldots \ldots
\[ \left( \frac{1}{2} \right)^{c \log n} \] chance of reaching height \( c \log n \)

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. Expect the right \( \# \) nodes/level. But could still get gaps.
Search path? \{ ~ \} to find $z$
Search path? ~

to find z

If p is a bend, we know that it was not promoted.
Otherwise we would have searched further in the level above.
Search path? to find $z$

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

The same holds for all nodes between $p$ and $q$. 
Search path? \{ \}

To find \( z \) \:

- \( \cdots \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow p \rightarrow \bullet \rightarrow Z \)

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.
Search path? to find $z$?

If $p$ is a bend, we know that it was not promoted.
Otherwise we would have searched further 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 50-50 chance of being promoted. $\Rightarrow$ segment length $= O(\log n)$

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

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

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

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

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

obtaining $\succ c \cdot \log n \text{ } H$ is very unlikely
Path from \( z \) to "start"

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

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

Because of previous analysis, \( H \) at high height implies many other consecutive \( H \) below.

\[ \text{obtaining } > c \log n \text{ } H \text{ is very unlikely} \]
Path from $z$ to "start"

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

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

Consider the odds of $\# \text{ total moves} > 10 \cdot c \log n \Rightarrow \text{ flip } 10 \cdot c \log n \text{ coins obtaining } \leq c \cdot \log n \text{ "H".}$
Path from $z$ to "start"

$\# \text{ of up moves} = \# \text{ levels} \leq c \cdot \log n \ \text{w.h.p.}

$\mathbb{Z}$ $\subseteq$ $\# \text{ total moves} \leq \# \text{ coin flips until getting c.log n \"H\".}$

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

$$\Pr \{ \exists c \log n \ H \} = \binom{10 \log n}{c \log n} \cdot \left( \frac{1}{2} \right)^{c \log n} \cdot \left( \frac{1}{2} \right)^{9 \log n} \Rightarrow \Pr. \text{ the rest are } T.$$
Path from $\xi$ to "start"

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

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

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

$$\Pr \{ \xi \leq c \cdot \log n \} = \binom{10 \cdot c \cdot \log n}{c \cdot \log n} \cdot \left( \frac{1}{2} \right)^{c \cdot \log n} \cdot \left( \frac{1}{2} \right)^{9 \cdot c \cdot \log n} \rightarrow \Pr. \text{ the rest are } T.$$
Flip $10 \cdot \log n$ coins.

$\Pr \{ \xi \leq \log n \} = {10 \log n \choose \log n} \cdot \left( \frac{1}{2} \right)^{9 \log n}$

$\left( \frac{y}{x} \right)^x \leq \left( e \frac{10 \log n}{\log n} \right)^{\log n} / 2^{9 \log n}$
Path from $z$ to "start"

$\text{# of UP moves} = \text{# levels} \leq c \cdot \log n \quad \text{whp}$

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

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

$P(r \leq c \log n H^3) \leq \binom{10 \log n}{c \log n} \cdot \left(\frac{1}{2}\right)^{9 \log n}$

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

$\leq \left(e^{\frac{10 \log n}{c \log n}}\right)^{c \log n} = 2^{\log_{10} e \cdot \log n} / 2^{9 \log n}$
Path from \( z \) to "start" 

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

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

Flip \( 10 \cdot c \log n \) coins.

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

\[ \left( \frac{y}{x} \right)^{y/x} \leq \left( e^{\frac{10 \log n}{\log n}} \right)^{\log n} \geq 2^{9 \log n} = 2^{\frac{1}{2} \cdot \log n \cdot 9} \leq \left( \frac{1}{2^{(9 - \log 10) \cdot \log n}} \right) \]
Flip $10 \cdot \log n$ coins.

$\Pr \{ \exists \leq \log n \text{ heads} \leq (10 \cdot \log n) \cdot (\frac{1}{2})^{9 \cdot \log n}$

Path from $z$ to "start"

# of UP moves = # levels $\leq c \cdot \log n$ WHP

$\leq \log 10e \cdot \log n$ / $2^{9 \cdot \log n}$ = $2^{(\log 10e - 9) \cdot \log n}$

$= \frac{1}{2^{(9 - \log 10e)} \cdot \log n}$

$(9 - \log 10e) \cdot c = \alpha$
Path from \( z \) to "start"

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

\[ z \leq \text{# total moves} \leq \text{# coin flips until getting} \]

\[ c \cdot \log n \text{ "H".} \]

Flip \( 10 \cdot c \log n \) coins.

\[ \Pr \{ \xi \leq c \log n \mid H^3 \} \leq \binom{10c \log n}{c \log n} \cdot \left(\frac{1}{2}\right)^9 c \log n \]

\[ (y) \leq (\frac{y}{x})^x \]

\[ \leq \left( e \frac{10c \log n}{c \log n} \right)^{c \log n} \cdot \frac{1}{2} \leq \frac{1}{2^{(9 - \log 10e) c \log n}} = \frac{1}{2^{9 - \log 10e}} \cdot \frac{1}{c \log n} = \frac{1}{n^\alpha} \]

\[ (9 - \log 10e) \cdot c = \alpha \]
Path from $z$ to "start"

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

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

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

$$Pr \{ \exists k \leq c \log n \text{ H} \} \leq \binom{10c \log n}{c \log n} \cdot \left(\frac{1}{2}\right)^{9c \log n}$$

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

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

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

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