INDICATOR RANDOM VARIABLES – GROUP TESTING

DISCRETE STRUCTURES II

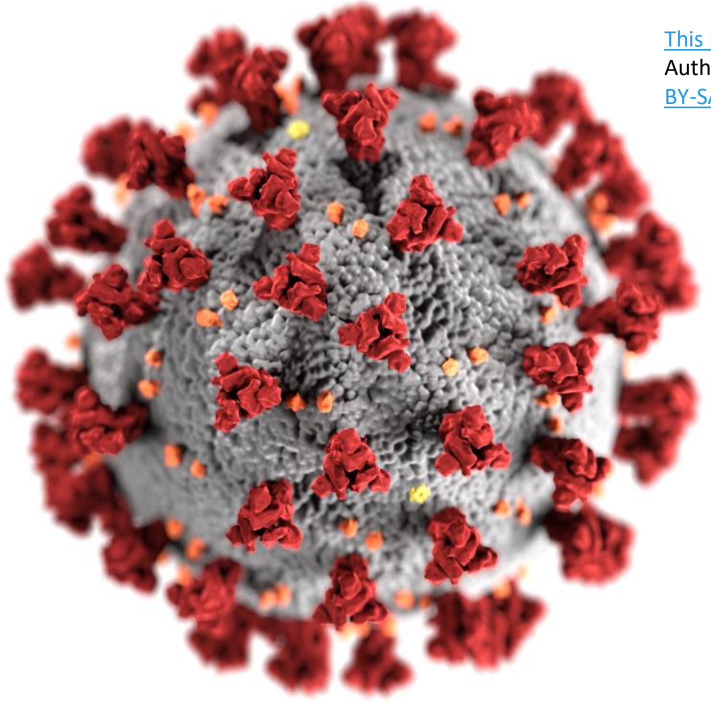
DARRYL HILL

BASED ON THE TEXTBOOK:

DISCRETE STRUCTURES FOR COMPUTER SCIENCE: COUNTING, RECURSION, AND PROBABILITY

BY MICHIEL SMID

Group Testing



COVID-19 test – gain a sample using a nasal swab.

All samples are taken to a lab to be analyzed.

Each sample analyzed costs time and money (and these days there are very many).

Robert Dorfman, 1943 – clever idea

Take all samples, divide into subgroups.

Combine all samples in a subgroup.

Use **one** test on the subgroup.

If negative: no one in the subgroup is infected.
If positive: at least one person in the subgroup has COVID.

To find out who, test everyone individually.

What is a good size for a subgroup?

n people $P_1, P_2, \ldots, P_n$

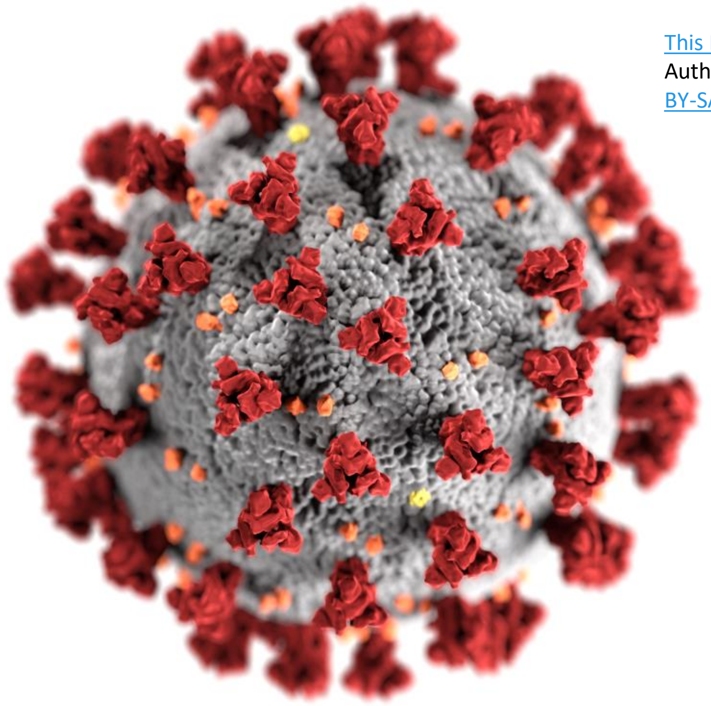
$k =$ # infected people, which we know

(we don't know k, but it should be possible to make a reasonable estimate)

We select a subset of the people to be tested for some pathogen (e.g. COVID-19), which we call T.

$$T \subseteq \{P_1, \ldots, P_n\}$$

We collect samples from every person in T. Take part of the samples from everyone in T and mix them together.



We then test the mixed sample of T with a single COVID-19 test. This test comes back positive for COVID-19, or negative for COVID-19.

n people $P_1, P_2, \ldots, P_n$

$k =$ # infected people, which we know

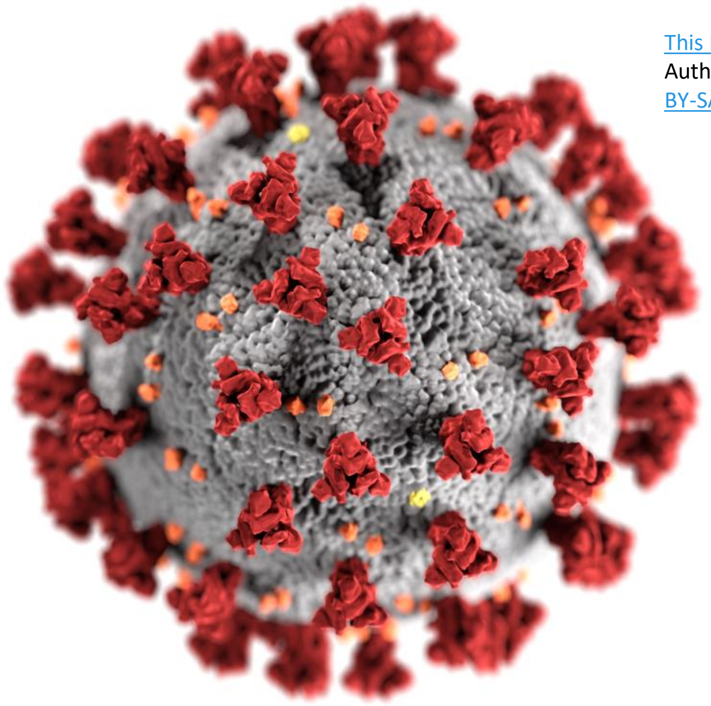
for $T \subseteq \{P_1, \ldots, P_n\}$: we Test($T$)

Test(T) = negative: no infection in T
Test(T) = positive: ≥ 1 infection in T

For 1 positive test to show up, $|T| \leq 64$
[COVID-19, Germany, Israel]

We will assume (unrealistically) that there are no false positives or false negatives.

(this will keep our analysis simpler)



If Test(T) = positive, then at least one person in T is infected. In this case we test everyone in T (from part of the sample that was set aside). $\sum_{P_i \in T} \mathrm{Test}(P_i)$

If Test(T) = negative, inform everyone in T that they are negative.

n people $P_1, P_2, \dots, P_n$

$k = $ # infected people, which we know

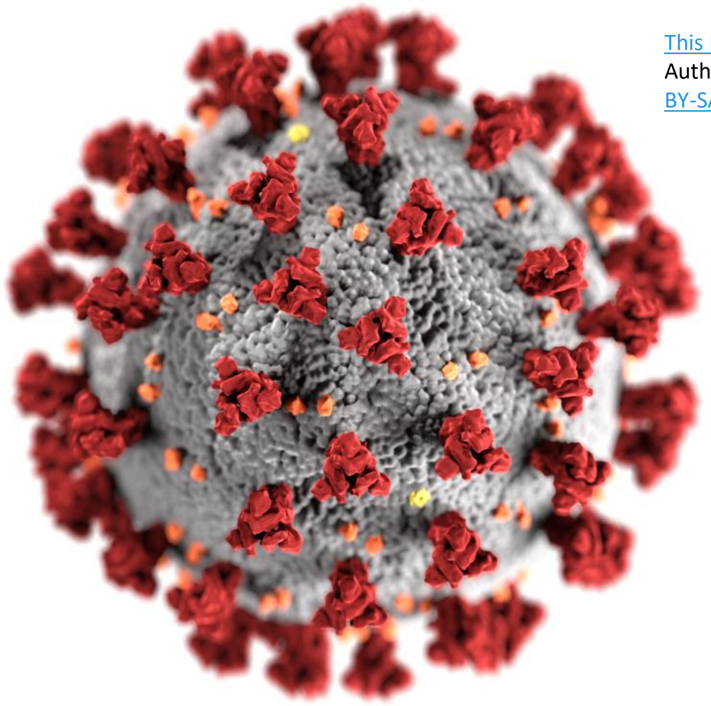
for $T \subseteq \{P_1, \dots, P_n\}$: we Test($T$)

Test(T) = negative: no infection in T
Test(T) = positive: ≥ 1 infection in T

For 1 positive test to show up, $|T| \leq 64$
[COVID-19, Germany, Israel]

We will assume (unrealistically) that there are no false positives or false negatives.

(this will keep our analysis simpler)



The goal is to use the fewest number of Tests (since they cost time and money).

In our model, smallest number of calls to Test().

n people $P_1, P_2, \dots, P_n$
for $T \subseteq \{P_1, \dots, P_n\}$
Test(T) = negative: no infection in T
Test(T) = positive: ≥ 1 infection in T

This method is known as *Single Pooling.*

We take n people and take a uniformly random permutation.

We divide this permutation into blocks of size s. For simplicity we assume s divides n evenly.

If it does not, then the last block is smaller, which will not greater affect the result.

Permutation of $P_1, P_2, \dots, P_n$:

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$				$\leftarrow s \rightarrow$
block 1	block 2				block $\frac{n}{s}$

We run Test on every block. So there are at least $\frac{n}{s}$ tests run, regardless of whether any test is positive or negative.

In addition, we may or may not run individual tests on the members of each block. We run tests on everyone if at least one person in the block has COVID. Since the individual tests may or may not happen, we will count them with indicator random variables.

n people $P_1, P_2, \ldots, P_n$
for $T \subseteq \{P_1, \ldots, P_n\}$
Test(T) = negative: no infection in T
Test(T) = positive: ≥ 1 infection in T

Permutation of $P_1, P_2, \ldots, P_n$:

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$				$\leftarrow s \rightarrow$
block 1	block 2		block j		block $\frac{n}{s}$

<u>SinglePooling(n, s)</u> // $\frac{n}{s}$ is an integer

uniformly random permutation of $P_1, \ldots, P_n$

for $j = 1, \ldots, \frac{n}{s}$:
    $t = $ Test(block j);
    if $t =$ negative:
        no infection in block j;
    if $t =$ positive:
        individual tests for j

$X =$ # calls to Test

There are at least $\frac{n}{s}$ tests, one for each block.

For the individual tests, we define Indicator Random Variables:

$$X_1, X_2, \ldots, X_n:$$

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

n people $P_1, P_2, \dots, P_n$
for $T \subseteq \{P_1, \dots, P_n\}$
Test(T) = negative: no infection in T
Test(T) = positive: ≥ 1 infection in T

SinglePooling(n, s) // $\frac{n}{s}$ is an integer

uniformly random permutation of $P_1, \dots, P_n$

for $j = 1, \dots, \frac{n}{s}$:
    t = Test(block j);
    if t = negative:
        no infection in block j;
    if t = positive:
        individual tests for j

Permutation of $P_1, P_2, \dots, P_n$:

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$				$\leftarrow s \rightarrow$
block 1	block 2		block j		block $\frac{n}{s}$

X = # calls to Test
$X_1, \dots, X_n$:

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

$$X = \frac{n}{s} + \sum_{i=1}^{n} X_i$$

$$E(X) = \frac{n}{s} + \sum_{i=1}^{n} E(X_i)$$

We need to determine $E(X_i)$

n people $P_1, P_2, \dots, P_n$
for $T \subseteq \{P_1, \dots, P_n\}$
Test(T) = negative: no infection in T
Test(T) = positive: ≥ 1 infection in T

X = # calls to Test, $X = X_1, \dots, X_n$:

$$X_i = \begin{cases} 1, \textit{if Test}(P_i) \textit{ is run} \\ 0, \textit{otherwise} \end{cases}$$

We need to determine $E(X_i)$

Assume P_i is in block j.

$Test(P_i)$ is run if and only if $Test(\text{block } j) = \text{positive}.$

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$				$\leftarrow s \rightarrow$
block 1	block 2		block j		block $\frac{n}{s}$

When is $Test(\text{block } j) = \text{positive?}$

There are two cases we should consider.

1. P_i has COVID.

2. P_i does not have COVID.

n people $P_1, P_2, \dots, P_n$
for $T \subseteq \{P_1, \dots, P_n\}$
Test(T) = negative: no infection in T
Test(T) = positive: ≥ 1 infection in T

X = # calls to Test, $X = X_1, \dots, X_n$:

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

We need to determine $E(X_i)$

Assume P_i is in block j.

$Test(P_i)$ is run if and only if $Test(\text{block } j) = \text{positive.}$

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$				$\leftarrow s \rightarrow$
block 1	block 2		block j		block $\frac{n}{s}$

When is $Test(\text{block } j) = \text{positive?}$

1. P_i has COVID.

In this case, $Test(\text{block } j) = \text{positive}$ (because of P_i)

Everyone in block j is tested, including P_i.

$\Pr(X_i = 1) = 1$, and thus:

$$E(X_i) = 1$$

n people $P_1, P_2, \ldots, P_n$
for $T \subseteq \{P_1, \ldots, P_n\}$
Test(T) = negative: no infection in T
Test(T) = positive: $\geq$ 1 infection in T

X = # calls to Test, $X = X_1, \ldots, X_n$:

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

We need to determine $E(X_i)$

Assume P_i is in block j.

$Test(P_i)$ is run if and only if $Test(\text{block } j) = \text{positive.}$

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$				$\leftarrow s \rightarrow$
block 1	block 2		block j		block $\frac{n}{s}$

When is $Test(\text{block } j) = \text{positive?}$

2. P_i does not have COVID.

In this case, $Test(\text{block } j) = \text{positive if}$ someone else in block j has COVID.

$$E(X_i) = \Pr(X_i = 1)$$

$= \Pr(\geq 1 \text{ other person infected in block j})$

n people $P_1, P_2, \dots, P_n$

for $T \subseteq \{P_1, \dots, P_n\}$

Test(T) = negative: no infection in T

Test(T) = positive: ≥ 1 infection in T

X = # calls to Test, $X = X_1 + \cdots + X_n$:

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

If P_i is infected: $E(X_i) = 1$.

If P_i is not infected: $E(X_i) = \Pr(X_i = 1)$

$= \Pr(\geq 1$ other person infected in P_i's block)

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$				$\leftarrow s \rightarrow$
block 1	block 2		block j		block $\frac{n}{s}$

Finding $\Pr(\geq 1$ other person infected in P_i's block) can be tricky, so we use the complement rule.

$\Pr((\geq 1$ infected in P_i's block))

$= 1 - \Pr(0$ infected in P_i's block)

Let $A = 0$ infected in P_i's block

Since it is a uniformly random permutation of the people, then $\Pr(A) = \frac{|A|}{|S|} = \frac{|A|}{n!}$.

$|A|$ = number of permutations where 0 infected in P_i's block

n people $P_1, P_2, \dots, P_n$
for $T \subseteq \{P_1, \dots, P_n\}$
Test(T) = negative: no infection in T
Test(T) = positive: ≥ 1 infection in T

X = # calls to Test, $X = X_1 + \cdots + X_n$:

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

If P_i is infected: $E(X_i) = 1$.

If P_i is not infected: $E(X_i) = \Pr(X_i = 1)$

$= \Pr(\geq 1 \text{ other person infected in } P_i\text{'s block})$

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$				$\leftarrow s \rightarrow$
block 1	block 2		block j		block $\frac{n}{s}$

Let $A = 0$ infected in P_i's block

$$\Pr(A) = \frac{|A|}{|S|} = \frac{|A|}{n!}.$$

$|A|$ = number of permutations where 0 are infected in P_i's block.

Can count this using the product rule.
Place P_i is one of n locations: n ways.

P_i is in block j let's say. P_i is not infected, now we must place $s - 1$ other (non-infected) people in block j, out of $n - k - 1$ uninfected people.

n people $P_1, P_2, \dots, P_n$
for $T \subseteq \{P_1, \dots, P_n\}$
Test(T) = negative: no infection in T
Test(T) = positive: ≥ 1 infection in T

X = # calls to Test, $X = X_1 + \cdots + X_n$:

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

If P_i is infected: $E(X_i) = 1$.

If P_i is not infected: $E(X_i) = \Pr(X_i = 1)$

$= \Pr(\geq 1$ other person infected in P_i's block)

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$				$\leftarrow s \rightarrow$
block 1	block 2		block j		block $\frac{n}{s}$

$|A|$ = number of permutations where 0 are infected in P_i's block.

Place P_i is one of n locations: n ways.

Choose $s-1$ non-infected people from $n-k-1$: $\binom{n-k-1}{s-1}$ ways.

Arrange these $s-1$ people: $(s-1)!$ ways.

There are $n-s$ people left to place: $(n-s)!$ ways.

$$n \cdot \binom{n-k-1}{s-1} \cdot (s-1)! \cdot (n-s)!$$

n people $P_1, P_2, \dots, P_n$
for $T \subseteq \{P_1, \dots, P_n\}$
Test(T) = negative: no infection in T
Test(T) = positive: ≥ 1 infection in T

X = # calls to Test, $X = X_1 + \cdots + X_n$:

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

If P_i is infected: $E(X_i) = 1$.

If P_i is not infected: $E(X_i) = \Pr(X_i = 1)$

$= \Pr(\geq 1 \text{ other person infected in } P_i\text{'s block})$

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$		P_i		$\leftarrow s \rightarrow$
block 1	block 2				block $\frac{n}{s}$

$$|A| = \left(n \cdot \binom{n-k-1}{s-1} \cdot (s-1)! \cdot (n-s)! \right)$$

$$= \frac{\binom{n-s}{k}}{\binom{n-1}{k}} \cdot n!$$

$$\Pr(A) = \frac{|A|}{|S|} = \frac{\binom{n-s}{k}}{\binom{n-1}{k}} \cdot n! \cdot \frac{1}{n!} = \frac{\binom{n-s}{k}}{\binom{n-1}{k}} = p$$

if P_i is infected, $E(X_i) = 1$
if P_i is not infected,

$$E(X_i) = 1 - \Pr(A) = 1 - \frac{\binom{n-s}{k}}{\binom{n-1}{k}} = 1 - p$$

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$		P_i		$\leftarrow s \rightarrow$
block 1	block 2				block $\frac{n}{s}$

X = # calls to Test

$$X = \frac{n}{s} + \sum_{i=1}^{n} X_i$$

$$E(X) = E\left(\frac{n}{s} + \sum_{i=1}^{n} X_i\right)$$

$$= E\left(\frac{n}{s}\right) + \sum_{i=1}^{n} E(X_i)$$

Event A = "no one in $P_i's$ block is infected"

$$\Pr(A) = \frac{\binom{n-s}{k}}{\binom{n-1}{k}} \cdot n! \cdot \frac{1}{n!} = \frac{\binom{n-s}{k}}{\binom{n-1}{k}} = p$$

If P_i is infected:

$X_i = 1$, which means $E(X_i) = 1$

If P_i is not infected:

$$E(X_i) = \Pr(X_i = 1 |\ P_i \text{ is not infected})$$
$$= 1 - \Pr(A) = 1 - p$$

For all $E(X_i)$ we differentiate between infected people and uninfected.

k people are infected: $E(X_i) = 1$

$n - k$ are uninfected: $E(X_i) = 1 - p$

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

Event A = "no one in $P_i' s$ block is infected"

$$\Pr(A) = \frac{\binom{n-s}{k}}{\binom{n-1}{k}} \cdot n! \cdot \frac{1}{n!} = \frac{\binom{n-s}{k}}{\binom{n-1}{k}} = p$$

If P_i is infected:

$X_i = 1$, which means $E(X_i) = 1$

If P_i is not infected:

$$E(X_i) = \Pr(X_i = 1 | \ P_i \text{ is not infected}) \\ = 1 - \Pr(A) = 1 - p$$

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$		P_i		$\leftarrow s \rightarrow$
block 1	block 2				block $\frac{n}{s}$

X = # calls to Test

$$E(X) = E\left(\frac{n}{s}\right) + \sum_{i=1}^{n} E(X_i)$$

$$= \frac{n}{s} + k \cdot 1 + (n-k) \cdot (1-p)$$

$$= \frac{n}{s} + k + (n-k) \cdot \left(1 - \frac{\binom{n-s}{k}}{\binom{n-1}{k}}\right)$$

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

Event A = "no one in $P_i's$ block is infected"

$$\Pr(A) = \frac{\binom{n-s}{k}}{\binom{n-1}{k}} \cdot n! \cdot \frac{1}{n!} = \frac{\binom{n-s}{k}}{\binom{n-1}{k}} = p$$

If P_i is infected:

$X_i = 1$, which means $E(X_i) = 1$

If P_i is not infected:

$$E(X_i) = \Pr(X_i = 1 |\ P_i \text{ is not infected}) \\ = 1 - \Pr(A) = 1 - p$$

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$		P_i		$\leftarrow s \rightarrow$
block 1	block 2				block $\frac{n}{s}$

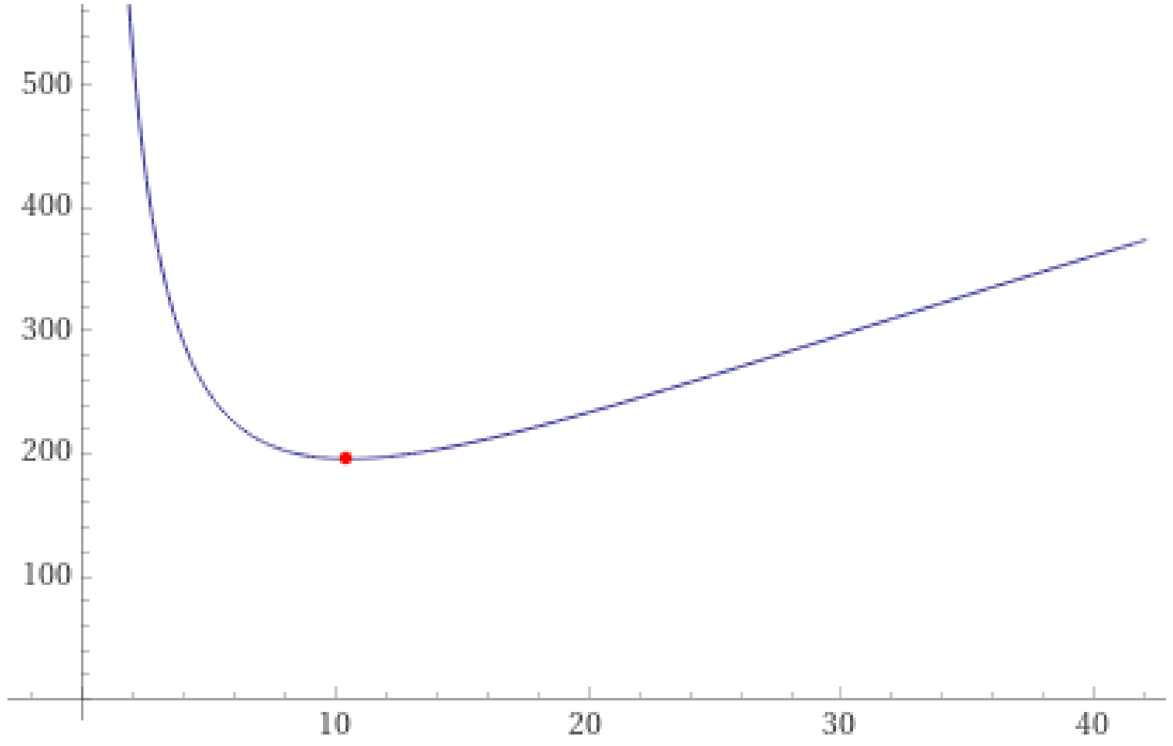
X = # calls to Test

$$E(X) = \frac{n}{s} + k + (n-k) \cdot \left(1 - \frac{\binom{n-s}{k}}{\binom{n-1}{k}}\right)$$

This is difficult to analyze by hand, but for given values of n and k we can plot it and find an approximate value for s that minimizes $E(X)$.

$$\min\left\{\frac{1000}{x} + 10 + 990\left(1 - \frac{\binom{1000-x}{10}}{\binom{999}{10}}\right) \middle| 0 \le x \le 40\right\} \approx 195.844 \text{ at } x \approx 10.4633$$

Plot:



(x from −2 to 42)

$$E(X) = \frac{n}{s} + k + (n-k) \cdot \left(1 - \frac{\binom{n-s}{k}}{\binom{n-1}{k}}\right)$$

If $n = 1000, k = 10$ then minimize $E(X)$ is at $s \approx 10$, where we run $\approx$ 196 tests

Using the naive method uses 1000 tests.

$k = 10$ is about 1% test positive.

MultiplePooling (n,c,s):

What we can do instead is to take a random permutation, as before, and divide into blocks as before.

But we repeat this process multiple (c) times. We take c random permutations, divide them into blocks, and test every block.

Then, for each P_i we examine the test results for all blocks that contained P_i.

If ≥ 1 is negative, then P_i must be negative.

1)

$\leftarrow s \rightarrow$				P_i	$\leftarrow s \rightarrow$
block 1				positive	block $\frac{n}{s}$

…

$\leftarrow s \rightarrow$		P_i			$\leftarrow s \rightarrow$
block 1		positive			block $\frac{n}{s}$

c)

$\leftarrow s \rightarrow$	P_i				$\leftarrow s \rightarrow$
block 1	negative				block $\frac{n}{s}$

If all c blocks containing P_i are positive, then P_i might be the reason they were all positive, so we test P_i individually.

MultiplePooling (n,c,s):

repeat c times:

uniformly random permutation of people

for $j = 1, \dots, \frac{n}{s}$: Test(block j)

for $i = 1, \dots, n$:

if ∃ iteration Test($P_i's$ block) = negative:

P_i not infected

else

Test(P_i);

1)

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$		P_i		$\leftarrow s \rightarrow$
block 1	block 2				block $\frac{n}{s}$

…

c)

$\leftarrow s \rightarrow$	$\leftarrow s \rightarrow$	P_i			$\leftarrow s \rightarrow$
block 1	block 2				block $\frac{n}{s}$

In SinglePooling, $c = 1$. So what happens when $c > 1$?

X = # Tests run

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

$$X = c \cdot \frac{n}{s} + \sum_{i=1}^{n} X_i$$

X = # Tests run

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

If P_i is infected, every single block that contains P_i tests positive for every permutation.

In this case, we are guaranteed to test P_i individually.

$X_i = 1$, and thus $E(X_i = 1)$.

1)

$\leftarrow s \rightarrow$				P_i	$\leftarrow s \rightarrow$
block 1				positive	block $\frac{n}{s}$

…

$\leftarrow s \rightarrow$		P_i			$\leftarrow s \rightarrow$
block 1		positive			block $\frac{n}{s}$

c)

$\leftarrow s \rightarrow$	P_i				$\leftarrow s \rightarrow$
block 1	positive				block $\frac{n}{s}$

$X = \#$ Tests run

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

If P_i is infected, $X_i = 1$

If P_i is not infected:

We test P_i if every block containing P_i on all c iterations has ≥ 1 infected person.

for $\ell = 1, \ldots, c$:
 $A_\ell =$ "iteration ℓ, no one in $P_i's$ block is infected“

Since this is easier to compute

$$X_i = 1 \leftrightarrow \overline{A_1} \wedge \overline{A_2} \wedge \cdots \wedge \overline{A_\ell}$$

$$E(X_i) = \Pr(X_i = 1)$$

$$\begin{aligned} &= \Pr(\overline{A_1} \wedge \overline{A_2} \wedge \cdots \wedge \overline{A_\ell}) \\ &= \Pr(\overline{A_1}) \cdot \Pr(\overline{A_2}) \cdot \ldots \cdot \Pr(\overline{A_\ell}) \\ &= [1 - \Pr(A_1)] \cdot \ldots \cdot [1 - \Pr(A_\ell)] \end{aligned}$$

We have that

$$\Pr(A_i) = \Pr(A) = \frac{\binom{n-s}{k}}{\binom{n-1}{k}} = p$$

(where A is the event from single pooling)

$X =$ # Tests run

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

If P_i is infected, $X_i = 1$

If P_i is not infected:

We test P_i if every block containing P_i on all c iterations has ≥ 1 infected person.

for $\ell = 1, .., c$:

$A_\ell =$ "iteration ℓ, no one in $P_i's$ block is infected"

Since this is easier to compute

$$X_i = 1 \leftrightarrow \overline{A_1} \wedge \overline{A_2} \wedge \cdots \wedge \overline{A_\ell}$$

$$E(X_i) = \Pr(X_i = 1)$$

$$\begin{aligned} &= \Pr(\overline{A_1} \wedge \overline{A_2} \wedge \cdots \wedge \overline{A_\ell}) \\ &= \Pr(\overline{A_1}) \cdot \Pr(\overline{A_2}) \cdot \ldots \cdot \Pr(\overline{A_\ell}) \\ &= [1 - \Pr(A_1)] \cdot \ldots \cdot [1 - \Pr(A_\ell)] \\ &= (1-p)^c \end{aligned}$$

If P_i is not infected, $E(X_i) = (1-p)^c$

$$E(X) = \frac{cn}{s} + \sum E(X_i)$$

When P_i is infected, $E(X_i) = 1$ (k times) and when P_i is not infected,

$$E(X_i) = (1-p)^c$$

$X = \#$ Tests run

$$X_i = \begin{cases} 1, if\ Test(P_i)\ is\ run \\ 0, otherwise \end{cases}$$

If P_i is infected, $X_i = 1$

If P_i is not infected:

We test P_i if every block containing P_i on all c iterations has ≥ 1 infected person.

for $\ell = 1, .., c$:
$A_\ell =$ "iteration ℓ, no one in $P_i' s$ block is infected"

Since this is easier to compute

$$X_i = 1 \leftrightarrow \overline{A_1} \wedge \overline{A_2} \wedge \cdots \wedge \overline{A_\ell}$$

$$E(X_i) = \Pr(X_i = 1)$$

$$\begin{aligned} &= \Pr(\overline{A_1} \wedge \overline{A_2} \wedge \cdots \wedge \overline{A_\ell}) \\ &= \Pr(\overline{A_1}) \cdot \Pr(\overline{A_2}) \cdot \ldots \cdot \Pr(\overline{A_\ell}) \\ &= [1 - \Pr(A_1)] \cdot \ldots \cdot [1 - \Pr(A_\ell)] \\ &= (1-p)^c \end{aligned}$$

If P_i is not infected, $E(X_i) = (1-p)^c$

$$E(X) = \frac{cn}{s} + 1 \cdot k + (n-k) \cdot (1-p)^c$$

$$= \frac{cn}{s} + 1 \cdot k + (n-k) \cdot \left(1 - \frac{\binom{n-s}{k}}{\binom{n-1}{k}}\right)^c$$

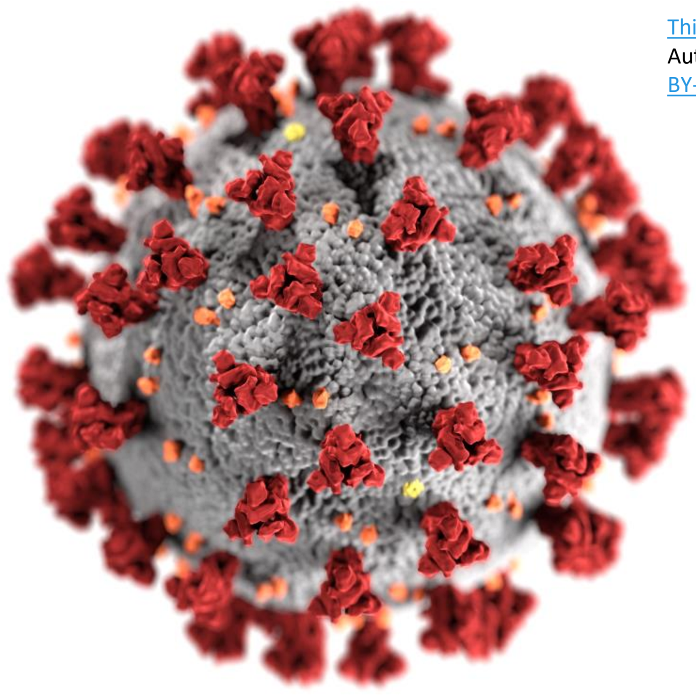
$$E(X) = \frac{cn}{s} + k + (n-k) \cdot \left(1 - \frac{\binom{n-s}{k}}{\binom{n-1}{k}}\right)^c$$

For $c = 1$, we have the SinglePooling expression

If $n = 1000, k = 10$:

$c = 1, s = 10$:	196 tests
$c = 2, s = 25$:	137 tests
$c = 3, s = 38$:	120 tests
$c = 4, s = 50$:	115 tests

This is (to my knowledge) not applied anywhere. There are perhaps practical limitations. For $c = 4$, you need to divide a sample in 5 parts.



Or the more complex a procedure, the greater the probability of error, and ruining a batch of tests.

Theoretical results are important, but are sometime impractical.

Skip Lists

Consider a set S of n numbers.

Data structure that supports:

SEARCH (x): Largest $w \in S$ s.t. $w \leq x$
INSERT (x): Inserts x into S
DELETE (x): Deletes x from S

Balanced binary search tree:

All three are $O(\log n)$ time.

Have to keep tree balanced through restructuring after insertion or deletion.

A skip list is constructed using outcomes of coinflips.

Balanced in expected sense (like QuickSort).

We define a sequence of lists: $S_0, S_1, S_2, \ldots$ which are subsets of S.

Let $S_0 = S$. Define a function flip $\in \{H, T\}$.

Let $i = 0$
For each $x \in S_i$:
  while $(\text{flip} = \text{H})$:
    add x to S_{i+1}
$i = i + 1$
Until $S_{i+1} = \emptyset$

Skip Lists

SEARCH (x): Largest $w \in S$ s.t. $w \leq x$
INSERT (x): Inserts x into S
DELETE (x): Deletes x from S

Let h = # non-empty sets above S_0

For each set S_i, construct a sorted linked list L_i.

Each u in L_i, key(u) = one element of S_i.

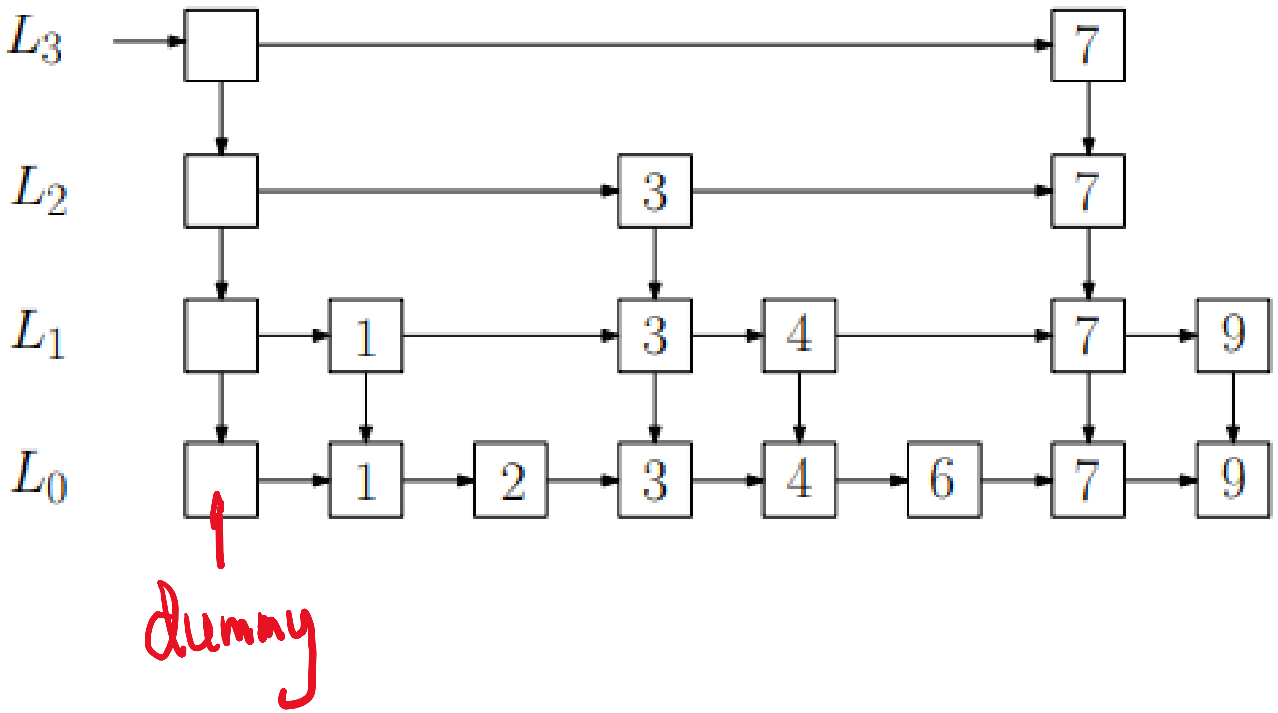
right(u)=successor node in of L_i (last node, right(u) = nil

down(u) = node in L_i points to the node with the same key in L_{i-1}

We add a dummy node to the beginning of each L_i. This is the *root.*

The resulting structure is a Skip List.

This example has height $h = 3$

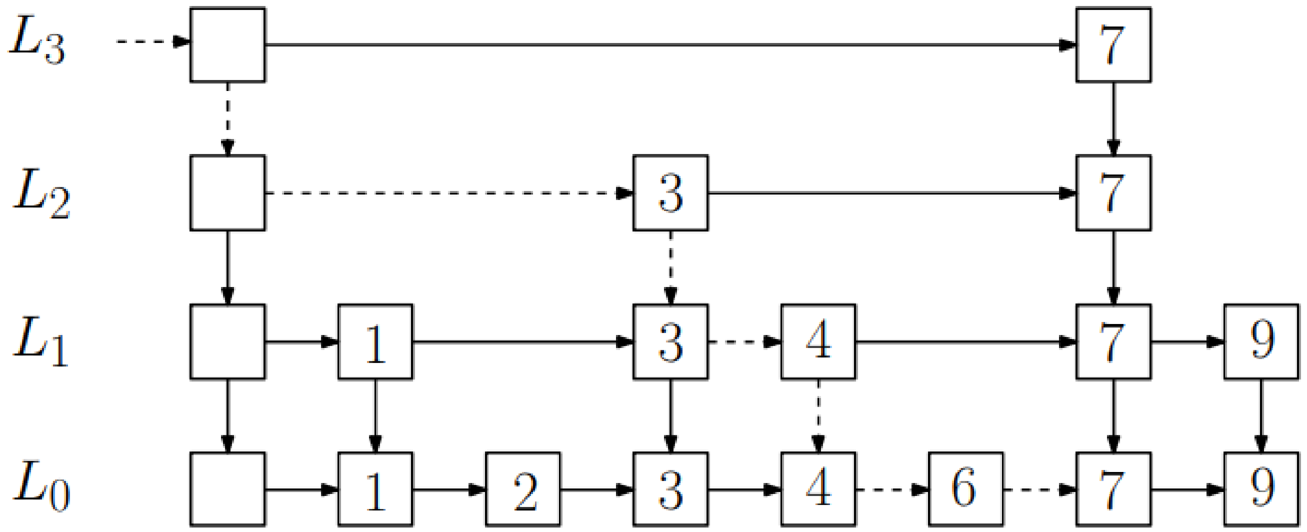


Skip Lists

SEARCH (x): Largest $w \in S$ s.t. $w \leq x$

```
u = root; i = h;
while i ≥ 1:
    if key(right(u)) < x
        u = right(u)
    else
        u = down(u)
        i = i − 1
//level 0
while key(right(u)) < x
        u = right(u)
```

Search path for 7:



Both INSERT and DELETE rely on SEARCH

Skip Lists

SEARCH (x): Largest $w \in S$ s.t. $w \leq x$

```
u = root; i = h;
while i ≥ 1:
    if key(right(u)) < x
        u = right(u)
    else
        u = down(u)
        i = i − 1
//level 0
while key(right(u)) < x
        u = right(u)
```

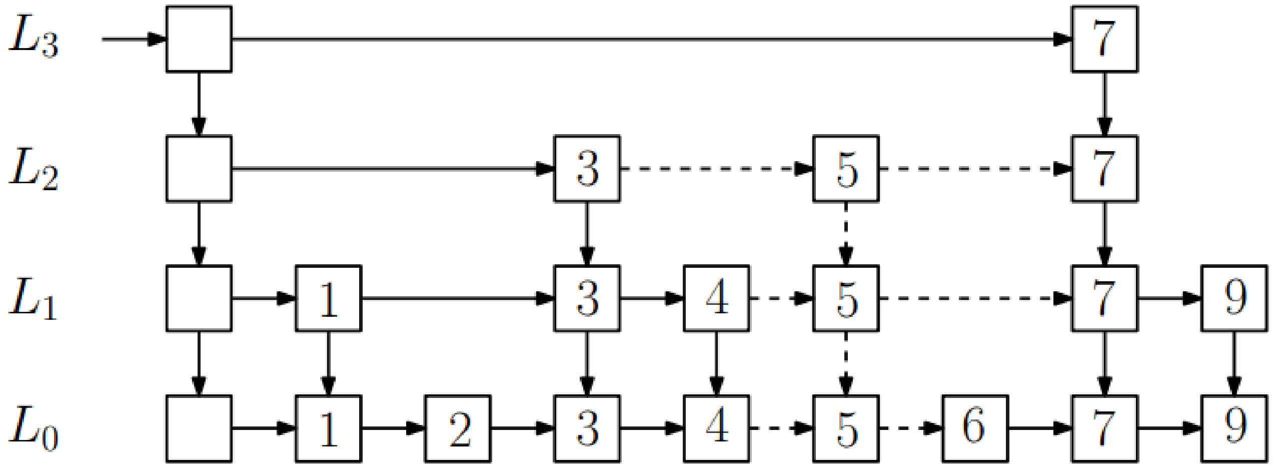
Both INSERT and DELETE rely on SEARCH

To insert a number, we do a search.

Once we find it's location, insert into L_0.

Then flip a coin. For each heads, insert into the next highest list.

Here is the insertion for 5.



Skip Lists

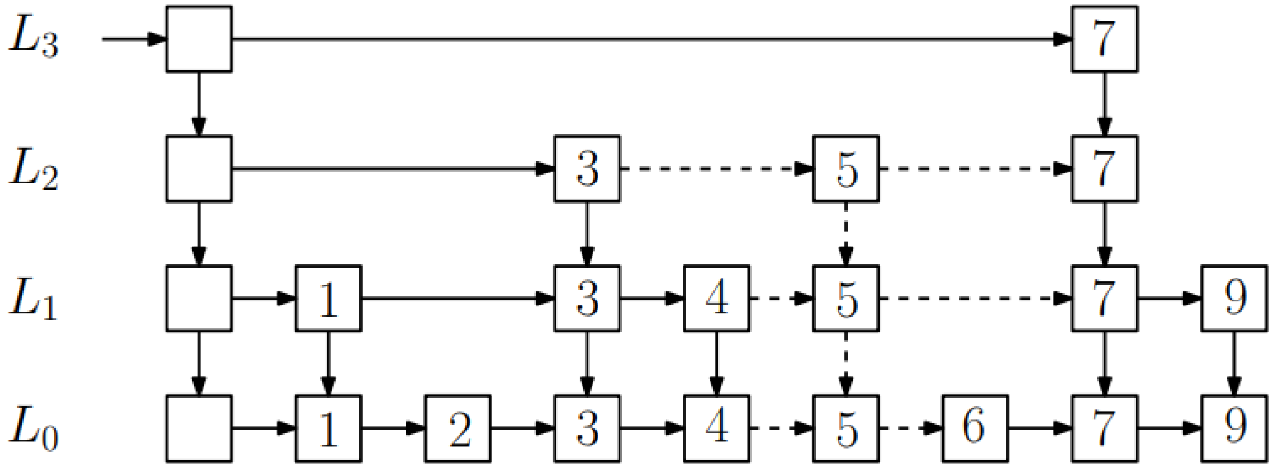
SEARCH (x): Largest $w \in S$ s.t. $w \leq x$

```
u = root; i = h;
while i ≥ 1:
    if key(right(u)) < x
        u = right(u)
    else
        u = down(u)
        i = i − 1
//level 0
while key(right(u)) < x
        u = right(u)
```

Both INSERT and DELETE rely on SEARCH

If the number of coin flips = heads is greater than h, the height of the dummy node is increased.

Easier to flip the coin beforehand.



Skip Lists

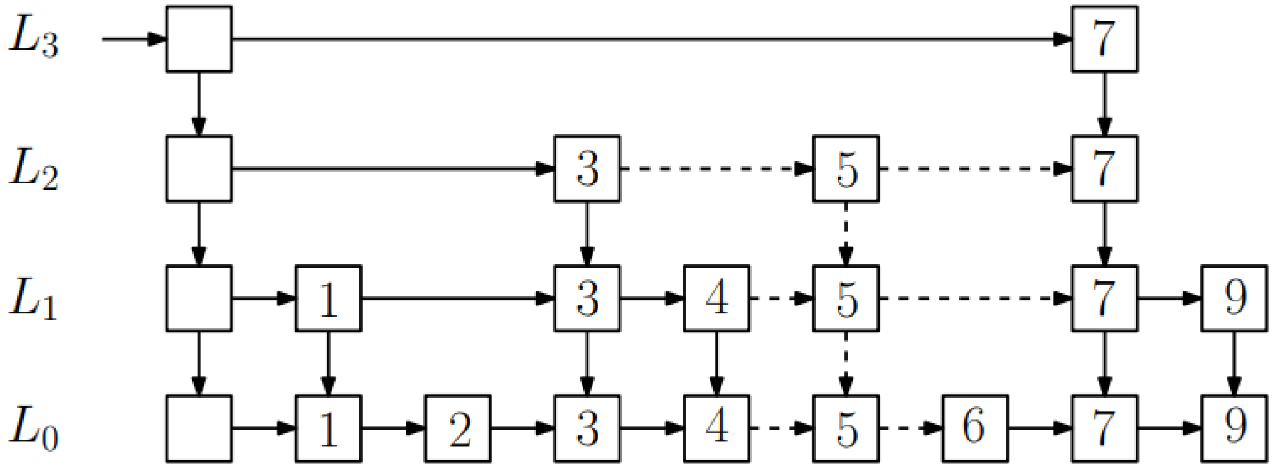
SEARCH (x): Largest $w \in S$ s.t. $w \leq x$

```
u = root; i = h;
while i ≥ 1:
    if key(right(u)) < x
        u = right(u)
    else
        u = down(u)
        i = i − 1
//level 0
while key(right(u)) < x
    u = right(u)
```

Both INSERT and DELETE rely on SEARCH

Delete is simply the opposite of search.

Once we find our number in L_i, we remove it from L_i and every list below it.



Skip Lists: Analysis

For any number x stored, what is

$$E\big(height(x)\big) = ?$$

This is equal to the expected number of coin flips until heads.

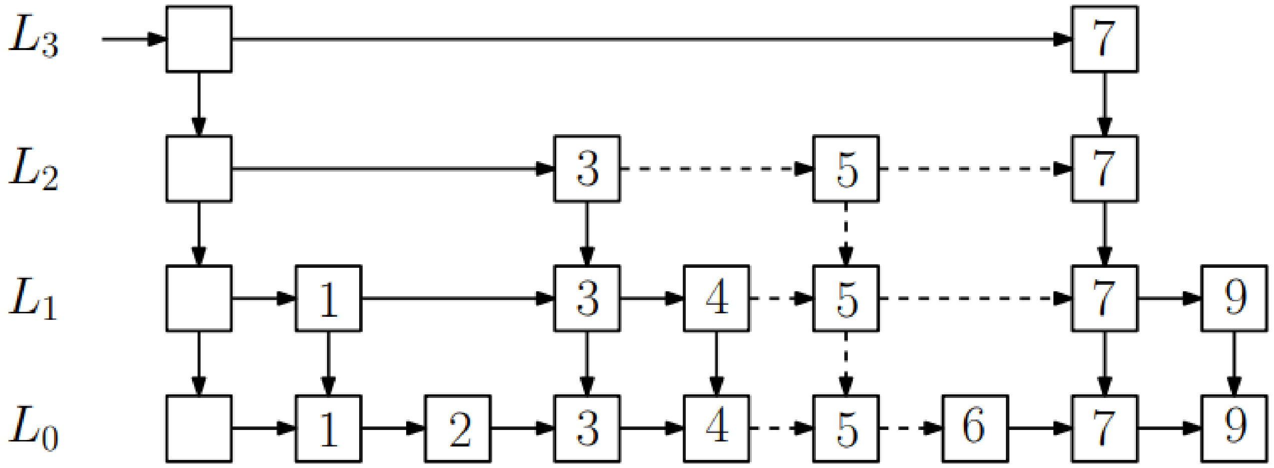
$$E\big(height(x)\big) = 1$$

(since we start at 0).

$$\Pr(x \in L_i) = ?$$

This requires i independent coin flips to come up tails. Thus

$$\Pr(x \in L_i) = \frac{1}{2^i}$$



Skip Lists: Analysis

What is $E(|L_i|)$?

We can use indicator random variables.

Let X be the number of elements in L_i.

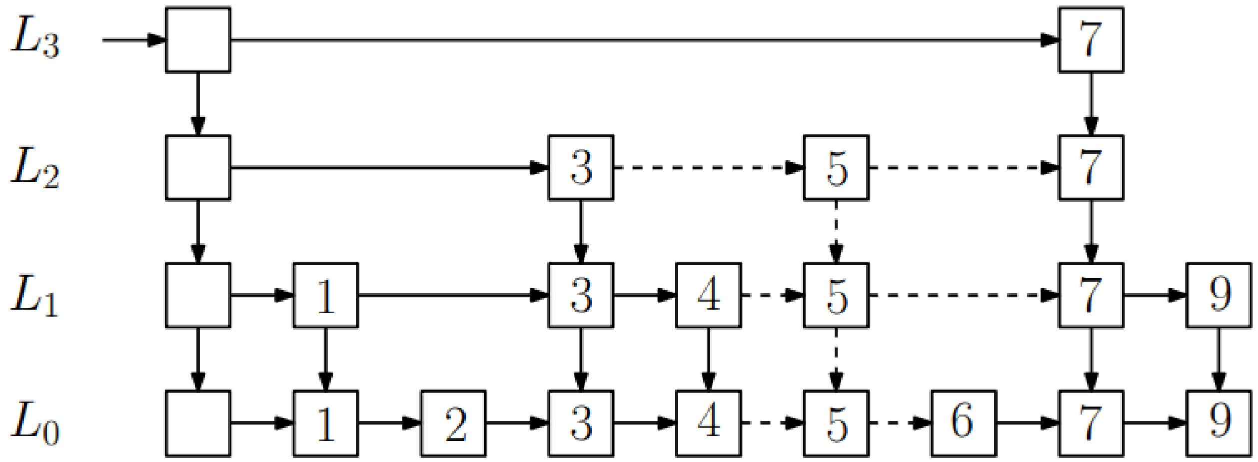
Let $X_1, X_2, \ldots, X_n$ be random variables where:

$$X_j = \begin{cases} 1 \text{ if the ith item from } L_0 \text{ is in } L_i \\ 0 \text{ otherwise} \end{cases}$$

$$E(X_j) =$$
$$\Pr(j \in L_i) = \frac{1}{2^i}$$

Using linearity of expectation:

$$X = X_1 + X_2 + \cdots + X_n$$
$$E(X) = E(X_1) + \cdots + E(X_n)$$
$$E(X) = \frac{n}{2^i}$$



Skip Lists: Analysis

Let X be the total number of nodes in the skip list (ignoring dummy nodes). What is $E(X)$?

$$X = \sum_x (1 + h(x))$$

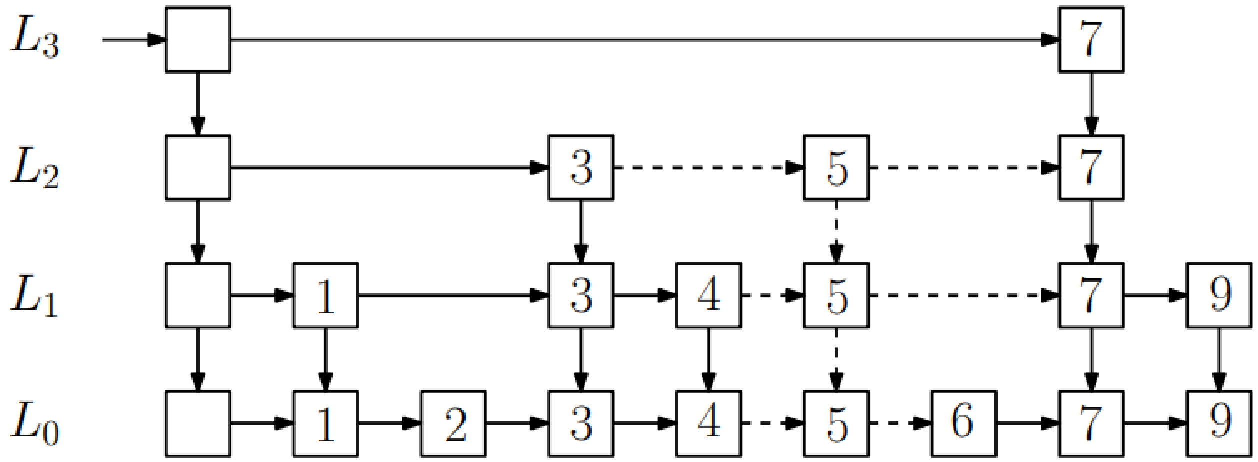
$$E(X) = E\left(\sum_x (1 + h(x))\right)$$

$$E(X) = \sum_x \big(1 + E\big(h(x)\big)\big)$$

$$E(X) = \sum_x 2 = 2n$$

Can also prove it as:

$$X = \sum_{i=0}^{h} |L_i|$$



Skip Lists: Analysis

What is $E(h)$?

Let $X_1, X_2, \ldots, X_n$ be random variables where:

$$X_i = \begin{cases} 1 \text{ if } L_i \text{ stores } \geq 1 \text{ number} \\ 0 \text{ otherwise} \end{cases}$$

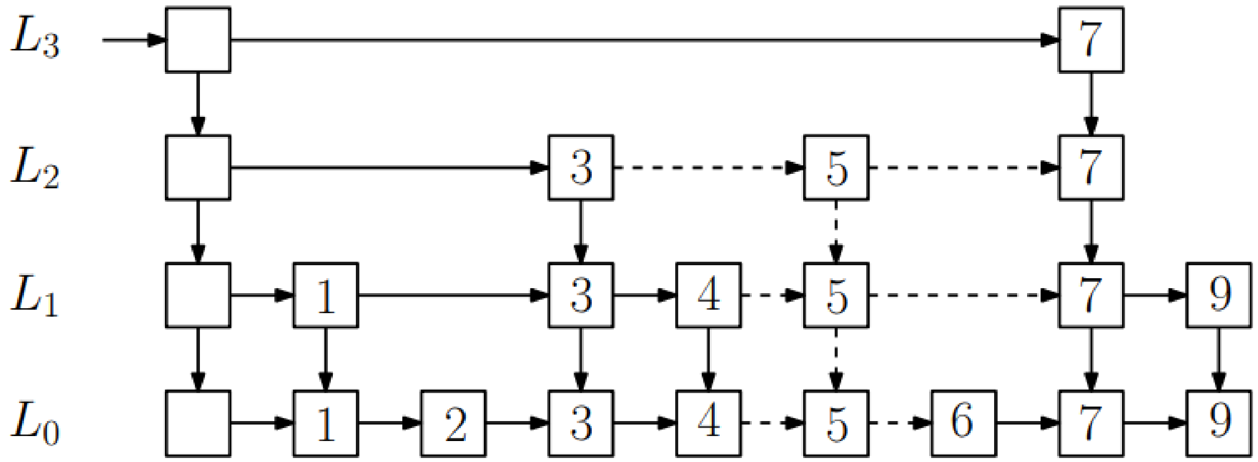
Then:

$$h = \sum_{i=0}^{\infty} X_i$$

It is obvious that $E(X_i) \leq 1$.

Also $X_i \leq |L_i|$, and thus

$$E(X_i) \leq E(|L_i|) \leq \frac{n}{2^i}$$



Skip Lists: Analysis

What is $E(h)$?

$$X_i = \begin{cases} 1 \text{ if } \ L_i \text{ stores } \ \geq 1 \text{ number} \\ 0 \text{ otherwise} \end{cases}$$

$E(X_i) \leq 1$ and $E(X_i) \leq \frac{n}{2^i}$

By linearity of expectation:

$$E(h) = \sum_{i=0}^{\infty} E(X_i)$$

$$E(h) = \sum_{i=0}^{\log n} E(X_i) + \sum_{i=\log n+1}^{\infty} E(X_i)$$

$$= \sum_{i=0}^{\log n} 1 + \sum_{i=\log n+1}^{\infty} \frac{n}{2^i}$$

$$= \log n + \sum_{j=0}^{\infty} \frac{n}{2^{\log n+1+j}}$$

$$= \log n + \sum_{j=0}^{\infty} \frac{n}{n \cdot 2^{1+j}}$$

Skip Lists: Analysis

What is $E(h)$?

$$X_i = \begin{cases} 1 \text{ if } \ L_i \text{ stores } \ \geq 1 \text{ number} \\ 0 \text{ otherwise} \end{cases}$$

$E(X_i) \leq 1$ and $E(X_i) \leq \frac{n}{2^i}$

By linearity of expectation:

$$E(h) = \sum_{i=0}^{\infty} E(X_i)$$

$$E(h) = \log n + \sum_{j=0}^{\infty} \frac{n}{n \cdot 2^{1+j}}$$

$$= \log n + \sum_{j=0}^{\infty} \frac{1}{2^{1+j}}$$

$$= \log n + \frac{1}{2} \sum_{j=0}^{\infty} \frac{1}{2^j}$$

$$= \log n + \frac{1}{2} \cdot 2$$

$$= \log n + 1$$

Skip Lists: Analysis

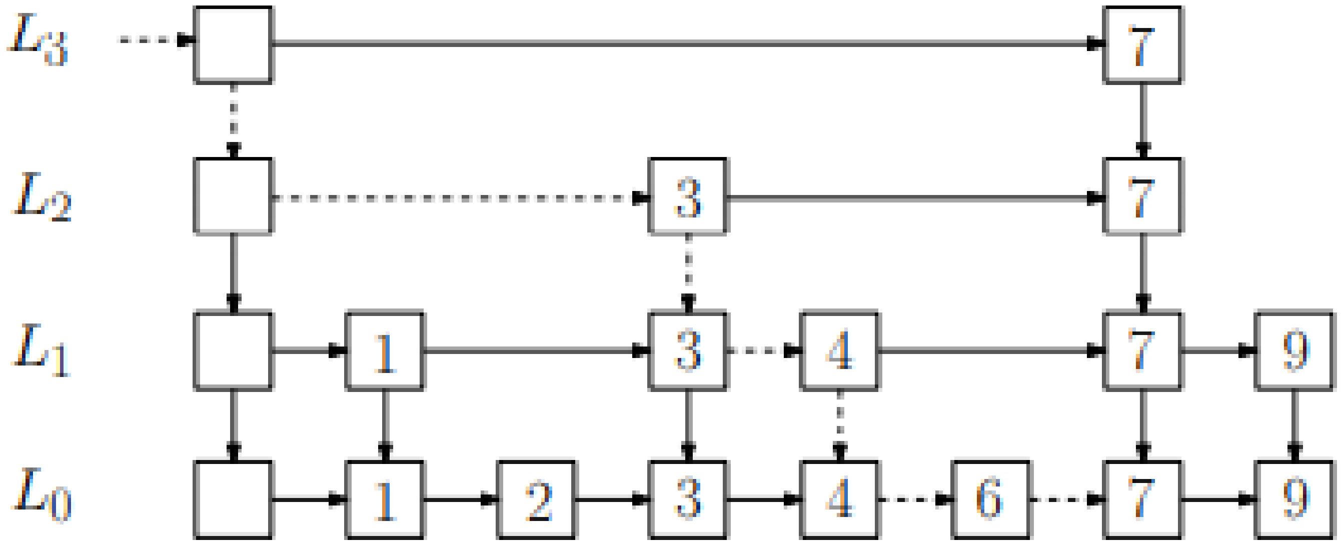
Since $E(h) = \log n + 1$, and the total number of dummy nodes is $h + 1$, then let Y be the expected number of nodes, including dummy nodes

$$Y = h + X + 1$$

$$E(Y) = E(h) + E(X) + E(1)$$

$$= 2n + \log n + 2$$

Next we bound the length of the search path.



Skip Lists: Analysis

Consider a number x in a node u and let v be the second last node on the search path to x.

Let N be the number of nodes on the search path to x.

Let M be the number of nodes on the path to v. Then $N = M + 1$.

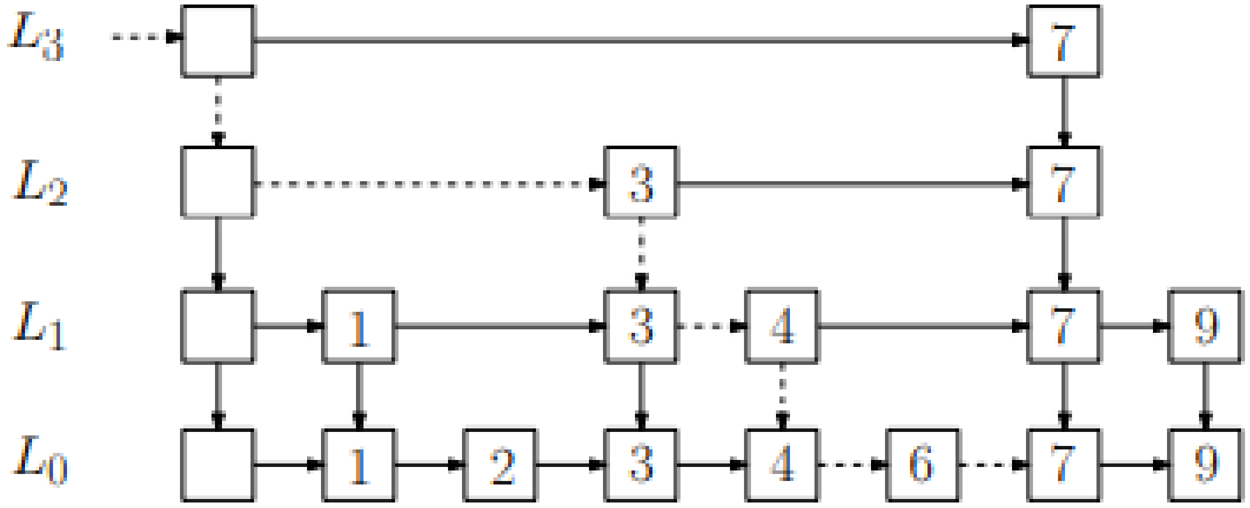
To simulate the path to v in reverse, start at v and

1. Walk up as far as you can
2. Walk left one step

The number of nodes on this reverse path is M.

Let M_i be the number of nodes in L_i where the reverse path walks left.

Then $M = h + 1 + \sum_{i=0}^{h} M_i$



Skip Lists: Analysis

Let M be the number of nodes on the path to v. Then $N = M + 1$.

Let M_i be the number of nodes in L_i where the reverse path walks left.

$$M = h + 1 + \sum_{i=0}^{h} M_i$$

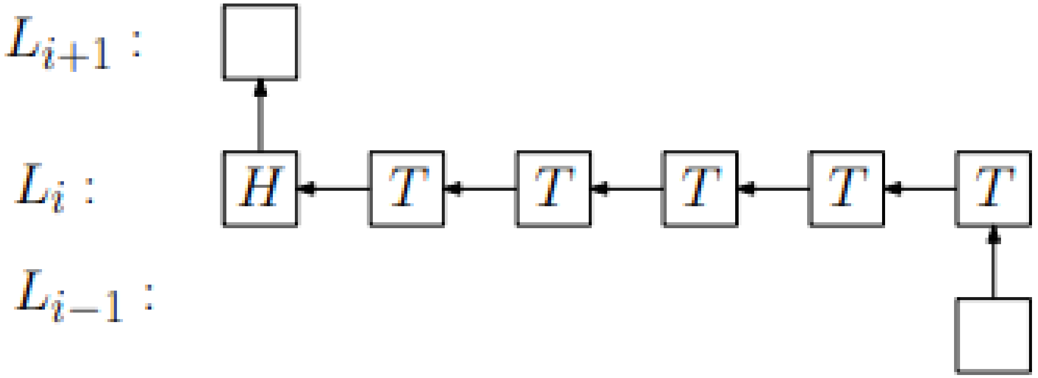
$$M = h + 1 + \sum_{i=0}^{\infty} M_i$$

$$E(M) = E(h) + 1 + \sum_{i=0}^{\infty} E(M_i)$$

M_i can be interpreted as the number of tails flipped until it comes up heads.

Thus $E(M_i) \leq 1$.

Also $E(M_i) \leq E(|L_i|) = \frac{n}{2^i}$

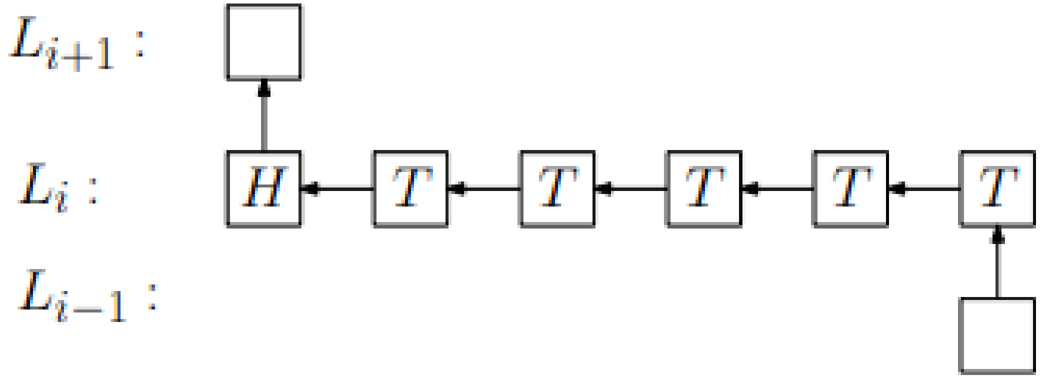


Skip Lists: Analysis

Let M be the number of nodes on the path to v. Then $N = M + 1$.

Let M_i be the number of nodes in L_i where the reverse path walks left.

$$E(M) = E(h) + 1 + \sum_{i=0}^{\infty} E(M_i)$$



$$= E(h) + 1 + \sum_{i=0}^{\log n} E(M_i) + \sum_{i=\log n+1}^{\infty} E(M_i)$$

$$= E(h) + 1 + \sum_{i=0}^{\log n} 1 + \sum_{i=\log n+1}^{\infty} \frac{n}{2^i}$$

$$= E(h) + 1 + \log n + \sum_{j=0}^{\infty} \frac{n}{2^{\log n+1+j}}$$

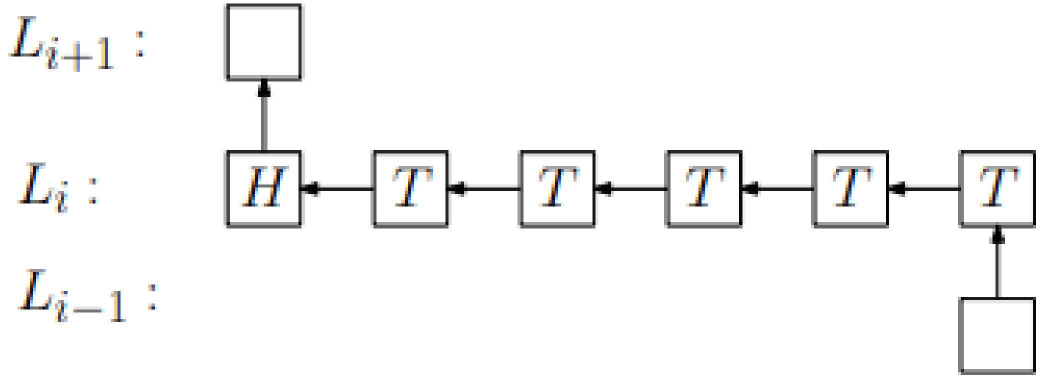
$$= E(h) + 1 + \log n + \sum_{j=0}^{\infty} \frac{n}{n \cdot 2^{1+j}}$$

Skip Lists: Analysis

Let M be the number of nodes on the path to v. Then $N = M + 1$.

Let M_i be the number of nodes in L_i where the reverse path walks left.

$$E(M) = E(h) + 1 + \sum_{i=0}^{\infty} E(M_i)$$



$$= E(h) + 1 + \log n + 1 + \sum_{j=0}^{\infty} \frac{n}{n \cdot 2^{1+j}}$$

$$= E(h) + 2 + \log n + \sum_{j=0}^{\infty} \frac{1}{2^{1+j}}$$

$$= E(h) + 2 + \log n + \frac{1}{2} \sum_{j=0}^{\infty} \frac{1}{2^j}$$

$$= \log n + 1 + 2 + \log n + \frac{1}{2} \cdot 2$$

$$= 2 \cdot \log n + 4$$

Skip Lists: Analysis

Let M be the number of nodes on the path to v. Then $N = M + 1$.

The expected length of a search path N is then

$$E(N) = E(M) + 1$$

$$= 2 \cdot \log n + 5$$

This bound also applies to INSERT and DELETE