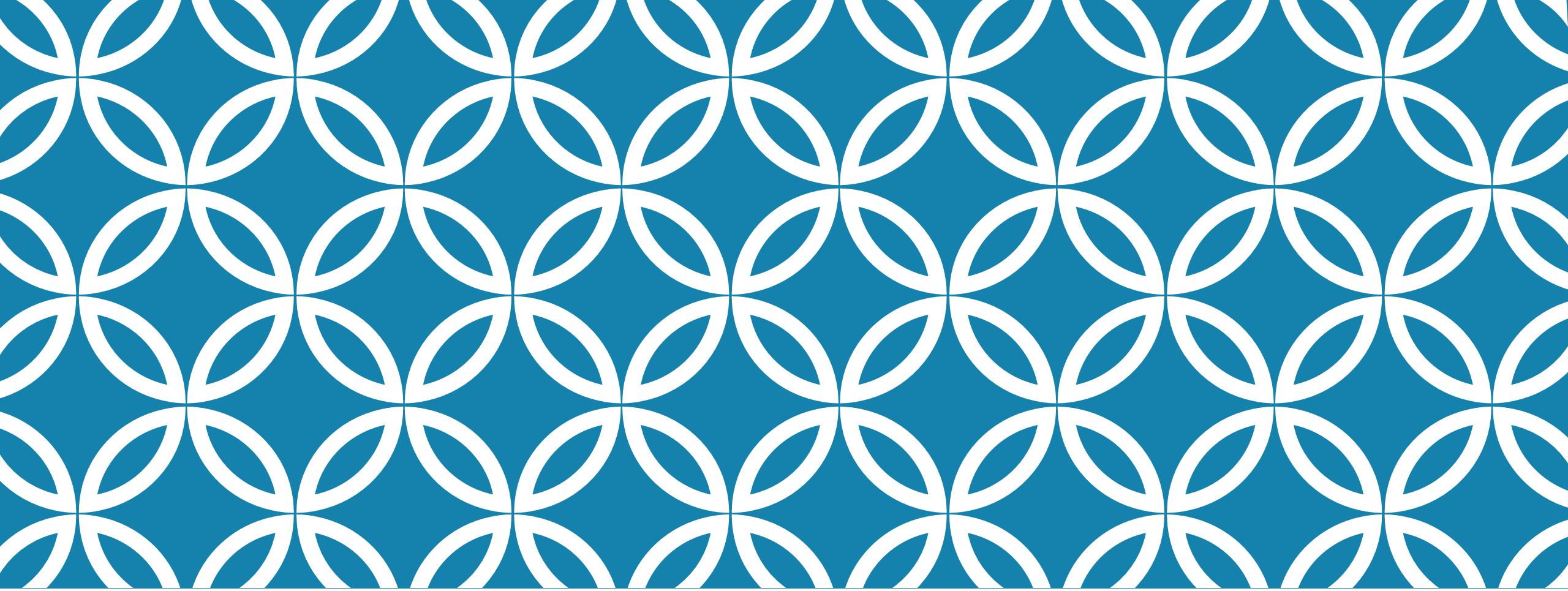
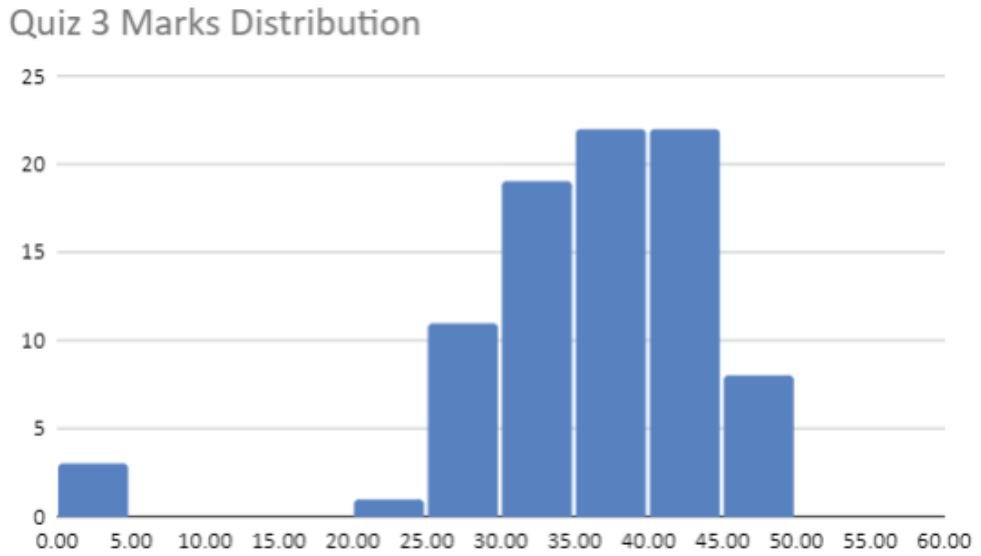


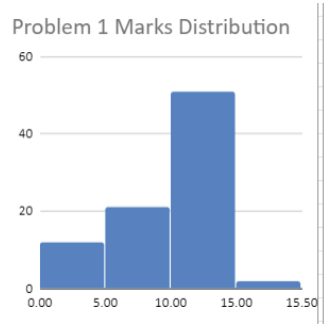
LECTURE 18: BLOOM FILTERS

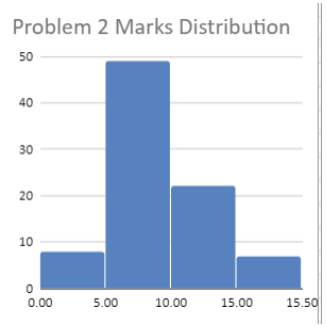
Harold Soh
harold@comp.nus.edu.sg

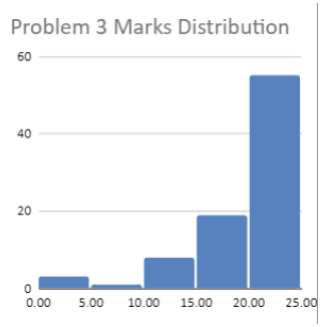
ADMINISTRATIVE ISSUES: QUIZ 3

- Finished grading Quiz 3
- Will be returned on Friday.
- More discussion tomorrow.









QUESTIONS?

Poll Everywhere

https://bit.ly/2LvG9bq

HASHING SUMMARY

Symbol Tables are *pervasive*

Hash tables are fast, efficient data structures for implementing symbol tables

- Under optimistic assumption, provably so.
- In the real world, often so.
- But not perfect!

Beats BSTs:

- for searching.
- but: gave up "order" operations.

CUCKOO HASHING

Did you know?

- Cuckoos lay eggs in other birds nests.
- When the cuckoo bird hatches, it pushes eggs/chicks out of the nest.

What a neat idea!

- Open addressing policy!
- Described by Rasmus Pagh and Flemming Friche Rodler in 2001.

CUCKOO HASHING: PERFORMANCE

Insertions seem to be quite complicated…

But: it takes expected $O(1)$ amortized time!

Analysis requires (a little) graph theory

To be continued in Week 12 …

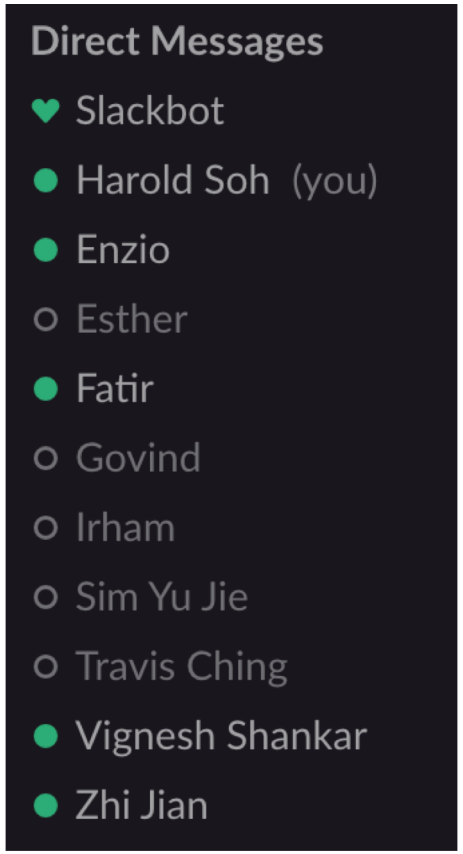
THE **NEXT 2 DAYS**

- Bloom Filters
- Tomorrow: Cuckoo Hashing

PROBLEM: CHECKING WHO IS ONLINE

I have a list of friends:

- Ayush
- Eldon
- Enzio
- Esther
- Fatir
- Govind
- Irham
- Ryan
- Si Jie
- Travis
- Vignesh
- Zhi Jian
- …



How can I quickly check who is online?

OTHER RELATED PROBLEMS

Spam filtering

- Have a list of valid emails and bad emails

Caching (avoiding spurious caching)

- Check if the a website is requested more than once

Preventing Denial-of-Service attacks

- Check if the request comes from a list of bad IP addresses

ONE SOLUTION?

Use a hash table!

Problem: size of the hash table can get large since we store the *keys*.

- Uses too much memory!

What if we had not enough memory?

0	0
1	0
2	www.gmail.com
3	www.apple.com
4	0
5	0
6	www.microsoft.com
7	0
8	www.nytimes.com
9	0

BLOOM FILTER

Space-efficient *Probabilistic* Data-Structure

A query returns:

- **True:** *Possibly* in set
- **False:** *Definitely* not in set

Elements can be added but not removed

- But we will deal with this later.

The more items, the larger the probability of **false positives**

- Saying True when the item is not in the set

FROM LECTURE 5: CORRECTNESS & EFFICIENCY

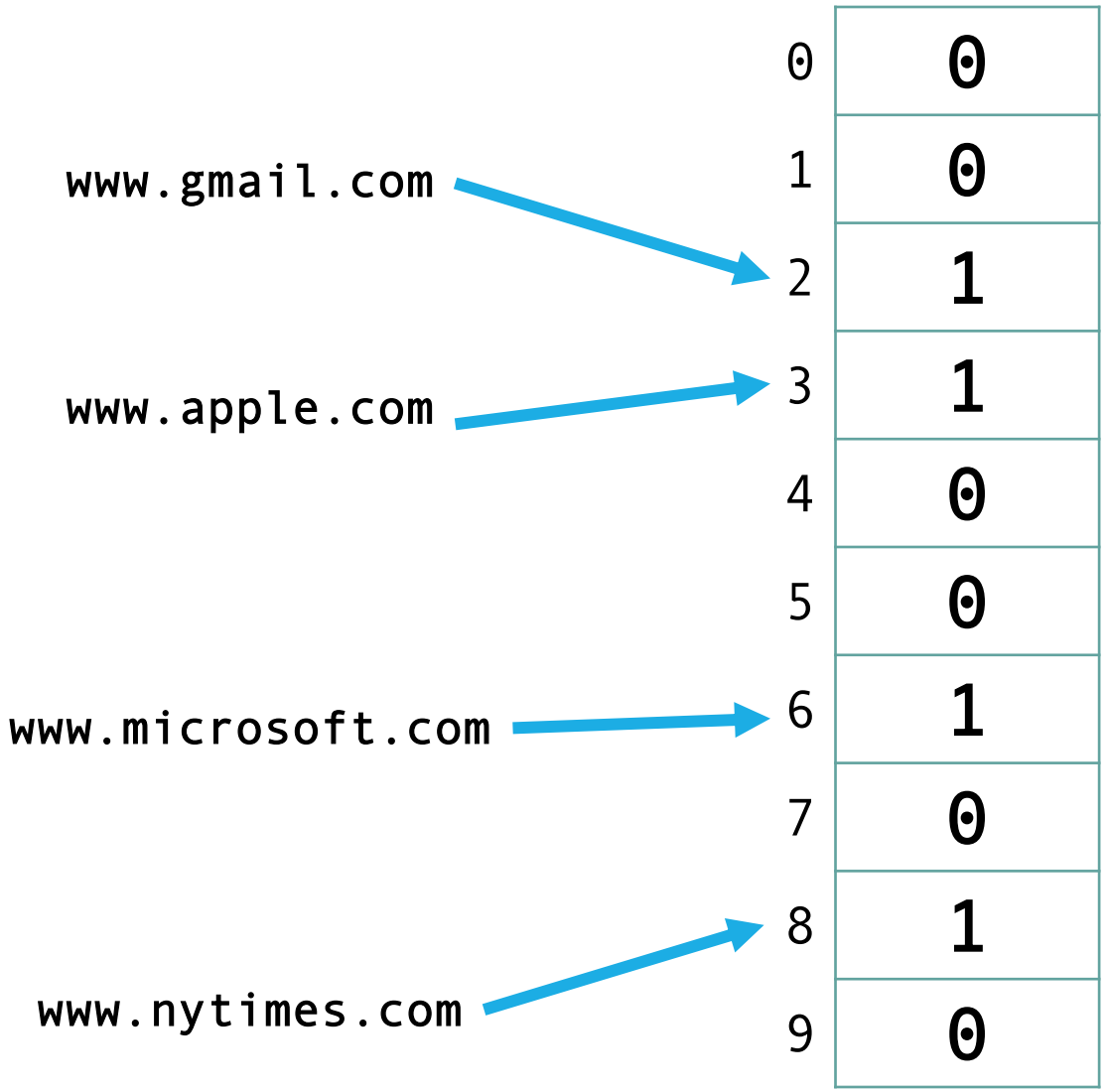
	Inefficient	Efficient
Incorrect	Slow & Wrong	Very fast… but wrong (sometimes useful)
Correct	Correct but slow (sometimes useful)	**We want this!**

IDEA: STORE LESS!

Only set k bits per item in a table of size m.

To start, let k = 1

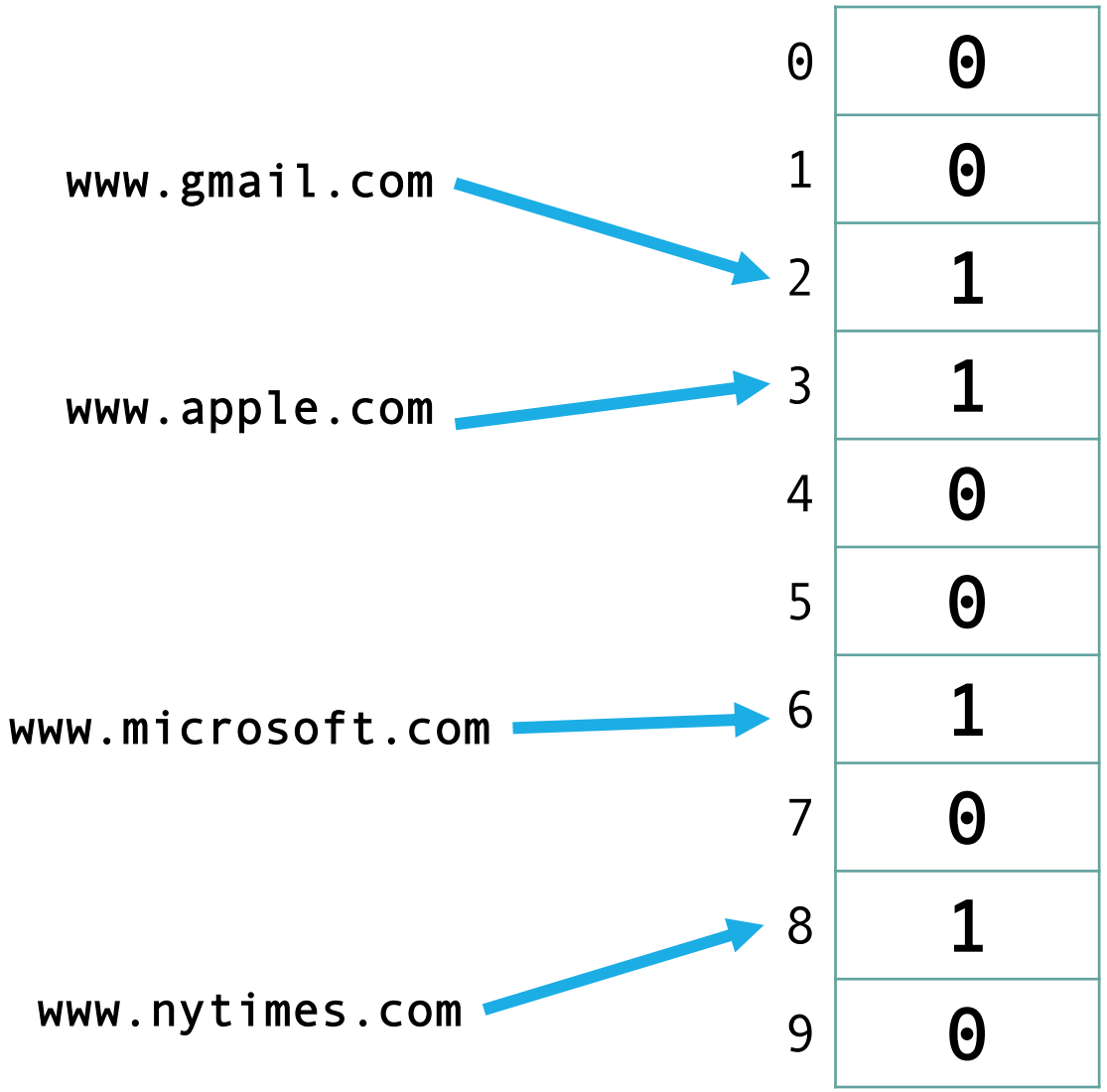
- For each item, we only set 1 bit!



IDEA: STORE LESS!

```
insert(key)
    h = hash(key)
    m_table[h] = 1

lookup(key)
    h = hash(key)
    return (m_table[h] == 1)
```

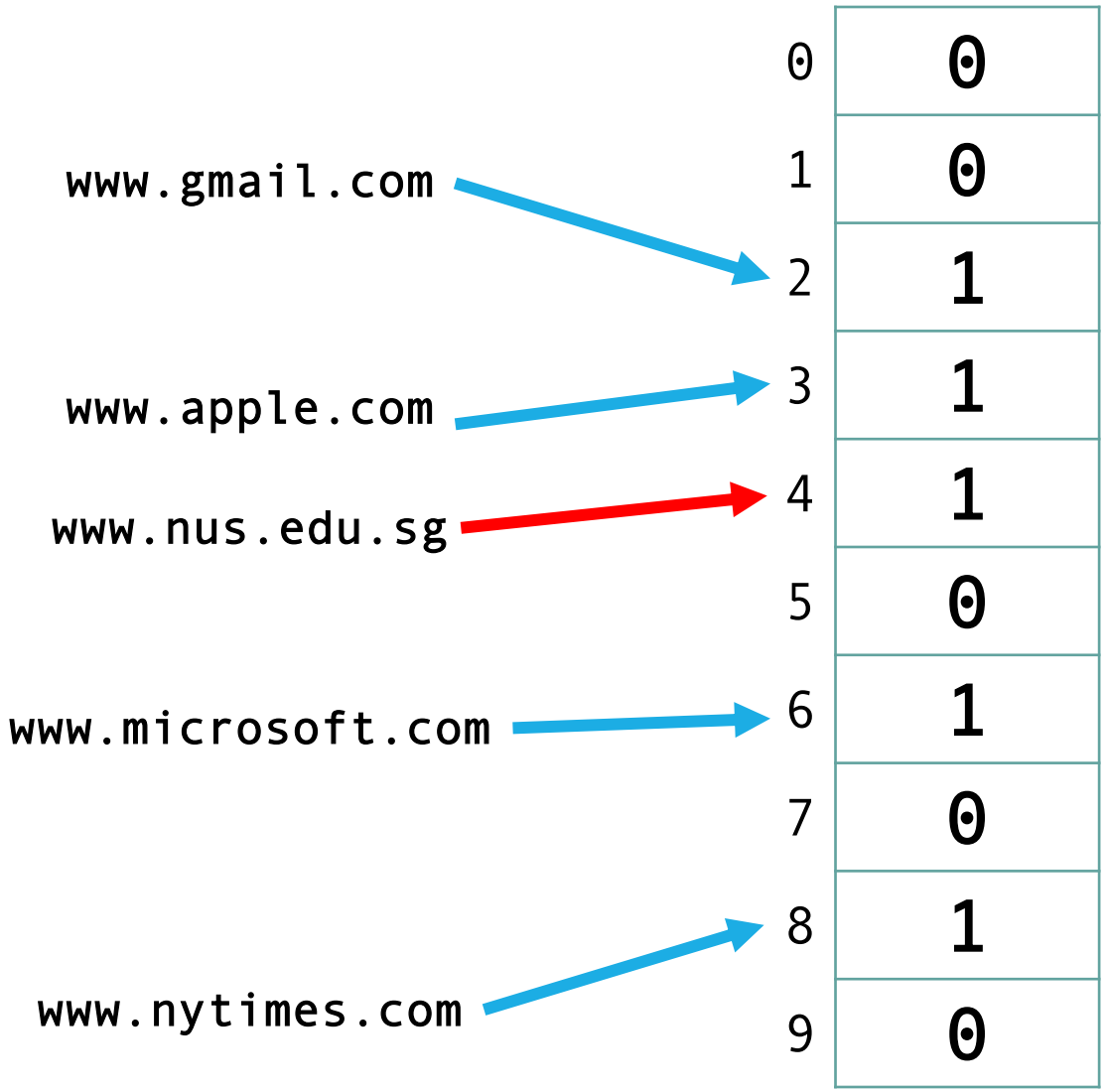


INSERTIONS

```
insert(key)
    h = hash(key)
    m_table[h] = 1
```

Insert www.nus.edu.sg

hash("www.nus.edu.sg") = 4



LOOKUPS

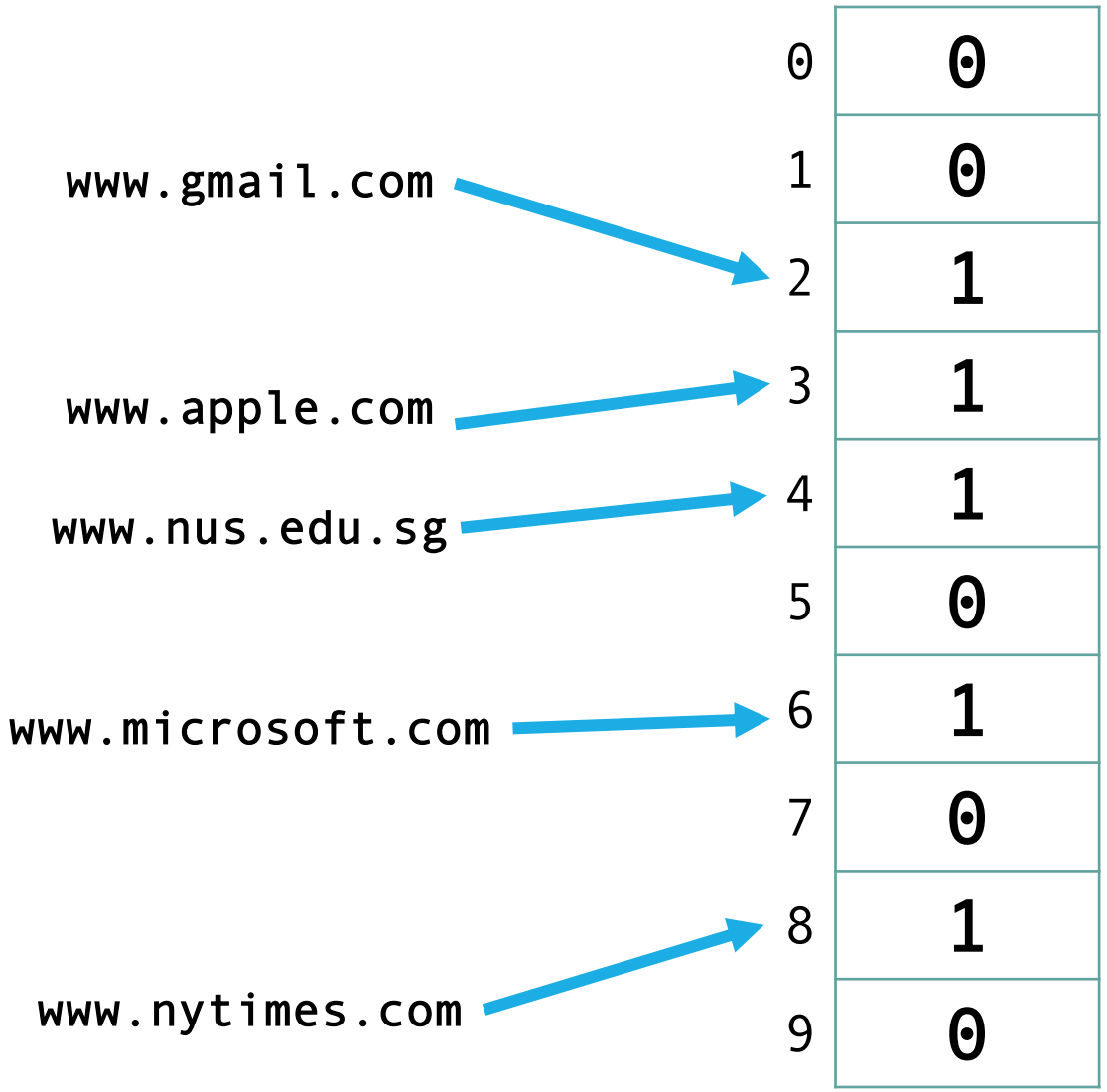
```
lookup(key)
    h = hash(key)
    return (m_table[h] == 1)
```

lookup www.nus.edu.sg

hash("www.nus.edu.sg") = 4

If key is in the table, will always return `true`.

No false negatives!



LOOKUPS

Returns `true` even ntu is not in the set!
FALSE POSITIVE due to collisions

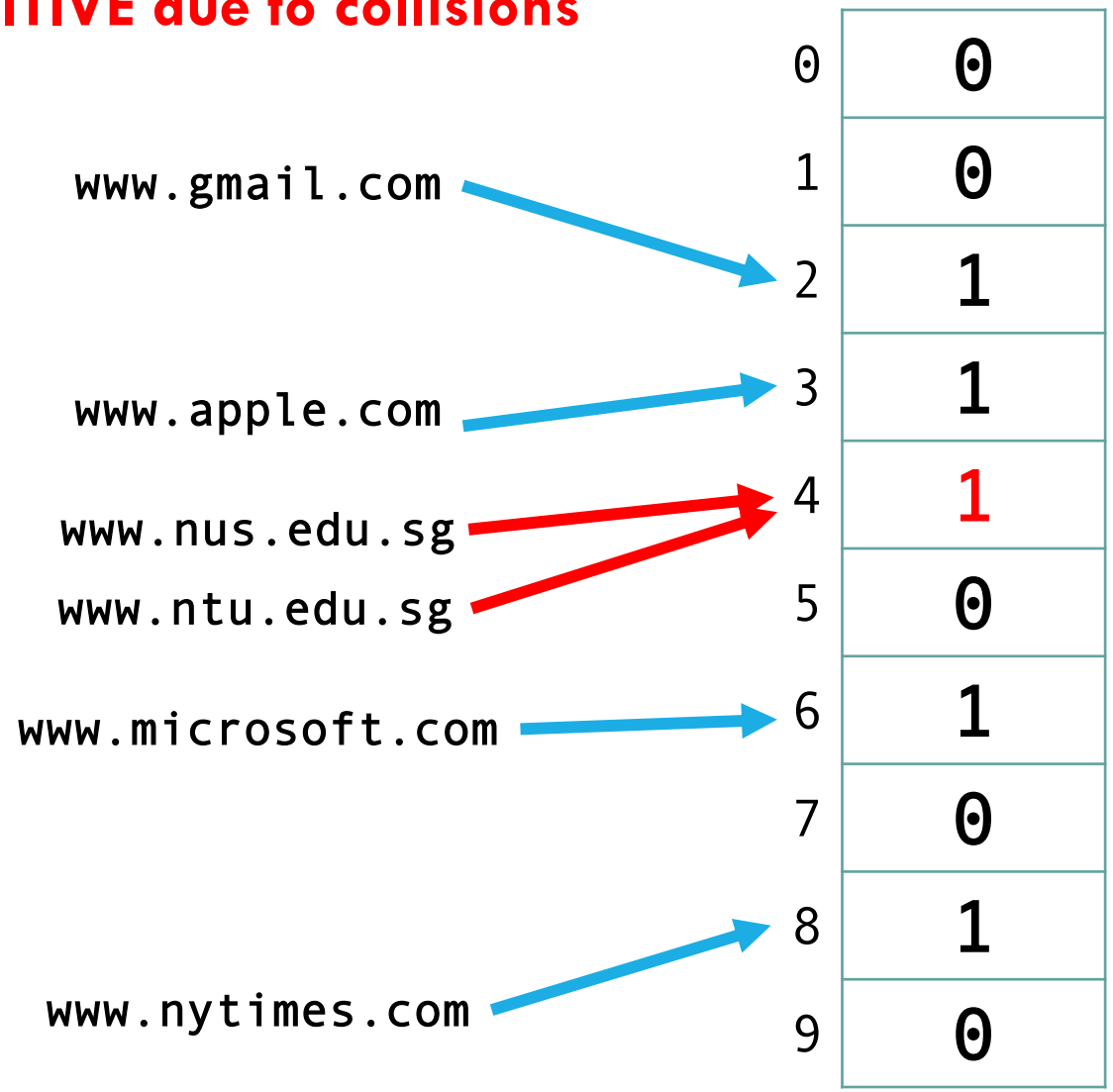
```
lookup(key)
    h = hash(key)
    return (m_table[h] == 1)
```

lookup www.nus.edu.sg

hash("www.nus.edu.sg") = 4

lookup www.ntu.edu.sg

hash("www.ntu.edu.sg") = 4



SPAM FILTERING

Given a Bloom Filter, would you store the good emails or bad emails?

A. Good Emails
B. Bad Emails
C. Not sure

SPAM FILTERING

Why? I would rather get an occasional spam message than put a real email (e.g., from my wife or boss) in the spam folder!

Given a Bloom Filter, would you store the good emails or bad emails?

A. Good Emails

B. Bad Emails

C. Not sure

PROBABILITY OF A FALSE POSITIVE?

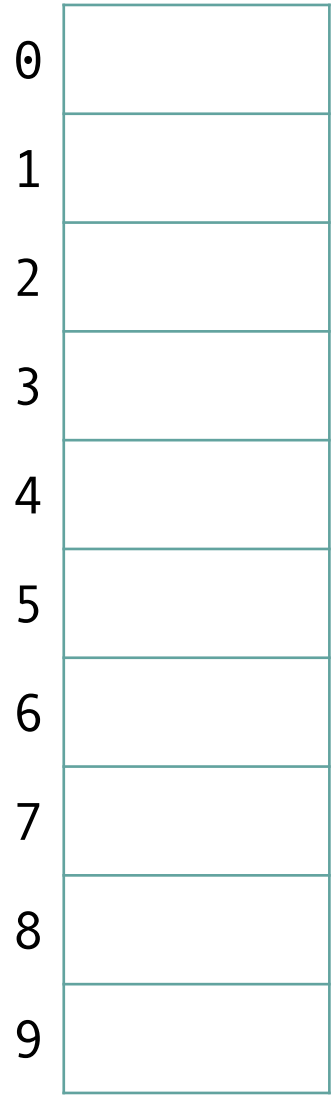
Table size is m, we store n elements

Set $k = 1$ bit per item

Test an item **not** in the set.

$$p(\text{false positive}) = 1 - p(\text{hits a cell with } 0)$$

$$p(\text{hits cell with } 0) = \underbrace{\left(1 - \frac{1}{m}\right)^n}_{\text{Elements are independently hashed}} \approx e^{-\frac{n}{m}}$$



THE NEXT 2 DAYS

- Probability Review
- Bloom Filters
- Cuckoo Hashing

PROBABILITY REVIEW

What do we **mean** when we say:

- "*the probability of getting an even number when rolling a die is ½*"
- "*the probability that I have the rare fatal disease is 90%*"

PROBLEM: YOU'RE NOT FEELING WELL...

You're not feeling well and go to the doctor.

You take a blood test.

Test comes back **positive** for *rare*, *fatal* disease.

Should you:

A. Skip CS2040S and start planning your demise?
B. Not worry.
C. Take the test again (and again) until it comes back negative.
D. Ask for more information.

PROBLEM: YOU'RE NOT FEELING WELL…

You're not feeling well and go to the doctor.

You take a blood test.

Test comes back **positive** for *rare*, *fatal* disease.

- Disease affects 0.1% of the population.
- Test correct identifies 99% of the people who have the disease.
- If you do not have the disease, test may come back positive 2% of the time.

What to do now?

Let's figure out the probability you actually have the disease!

PROBABILITY SPACE

A probability space (Ω, E, P) models a process consisting of outcomes that occur randomly.

Consists of three parts:

- Outcome or sample space Ω
- Event space E
- Probability function $P: E \to \mathbb{R}$

OUTCOME SPACE

Outcome space is a space of possible outcomes or "elementary outcomes", denoted by Ω.

Example: Outcomes of a dice roll, $\Omega = \{1,2,3,4,5,6\}$.

EVENT SPACE

Event space $E \subseteq 2^{\Omega}$ is a subset of the power set of Ω, it is the set of measurable events to which we assign probabilities.

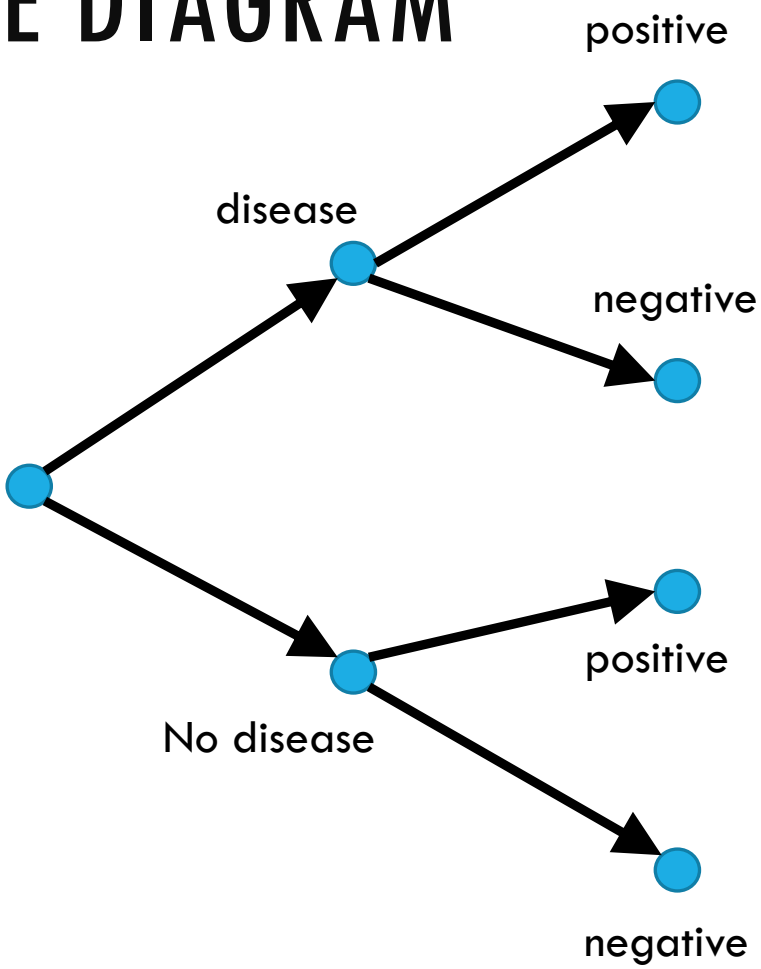
Example: The event space on whether a dice roll is odd or even, $E = \{\emptyset, \{1,3,5\}, \{2,4,6\}, \Omega\}$.

EVENT SPACE

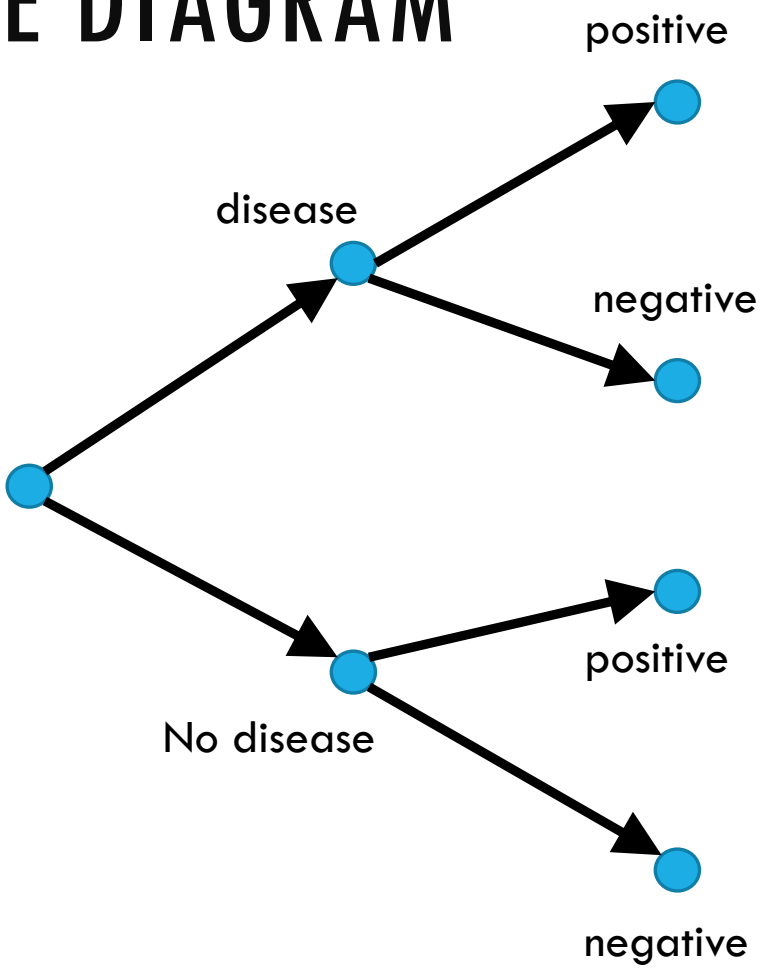
Event space must satisfy three basic properties:

1. It contains the empty event $\emptyset$, and the trivial event Ω.
2. It is closed under union, i.e. if $\alpha, \beta \in E$, then so is $\alpha \cup \beta$.
3. It is closed under complement, i.e. if $\alpha \in E$, then so is $\Omega - \alpha$.

TREE DIAGRAM



TREE DIAGRAM



Outcome Space Ω

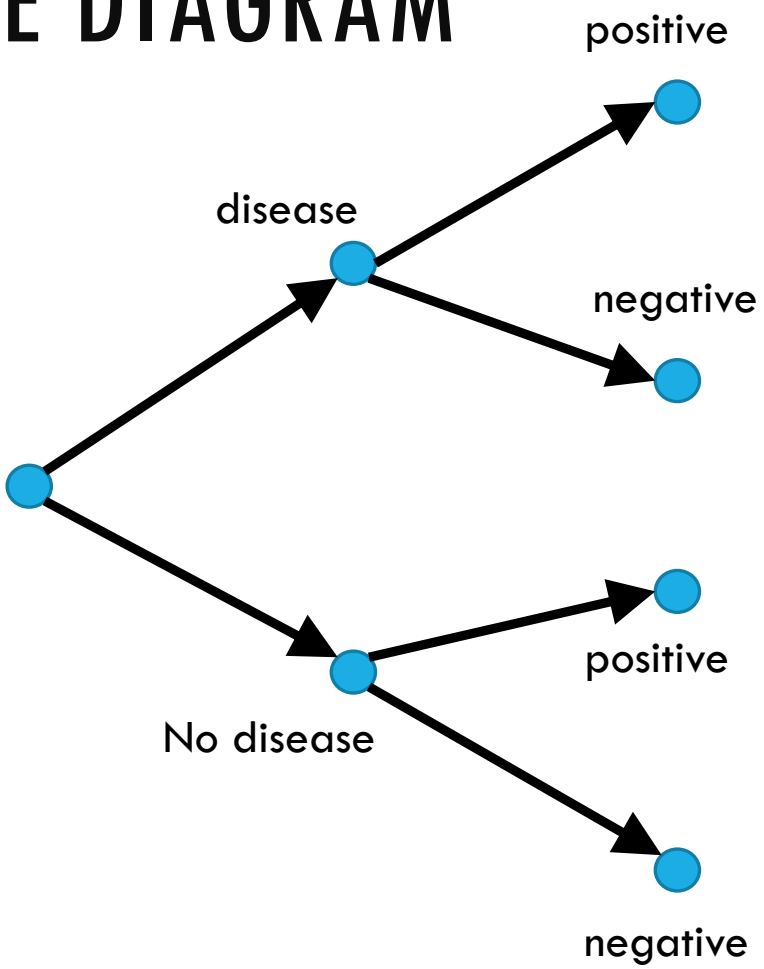
(disease, positive)

(disease, negative)

(no disease, positive)

(no disease, negative)

TREE DIAGRAM



Outcome Space Ω

(disease, positive)

(disease, negative)

(no disease, positive)

(no disease, negative)

Example Event Spaces:

Test is positive or negative

Event Space E

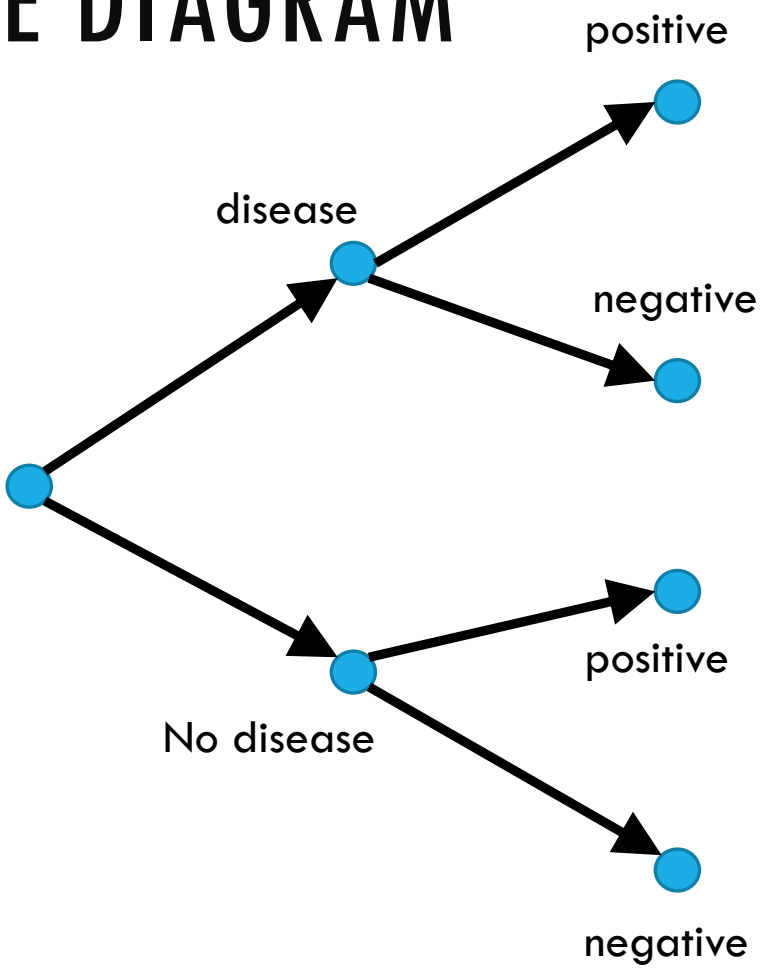
$\emptyset$

{(disease, positive),
(no disease, positive)}

{(disease, negative),
(no disease, negative)}

Ω

TREE DIAGRAM



Outcome Space Ω

(disease, positive)

(disease, negative)

(no disease, positive)

(no disease, negative)

Example Event Spaces:

Test is positive or negative

Event Space E

$\emptyset$

{(disease, positive),
(no disease, positive)}

{(disease, negative),
(no disease, negative)}

Ω

Disease or no disease

Event Space E

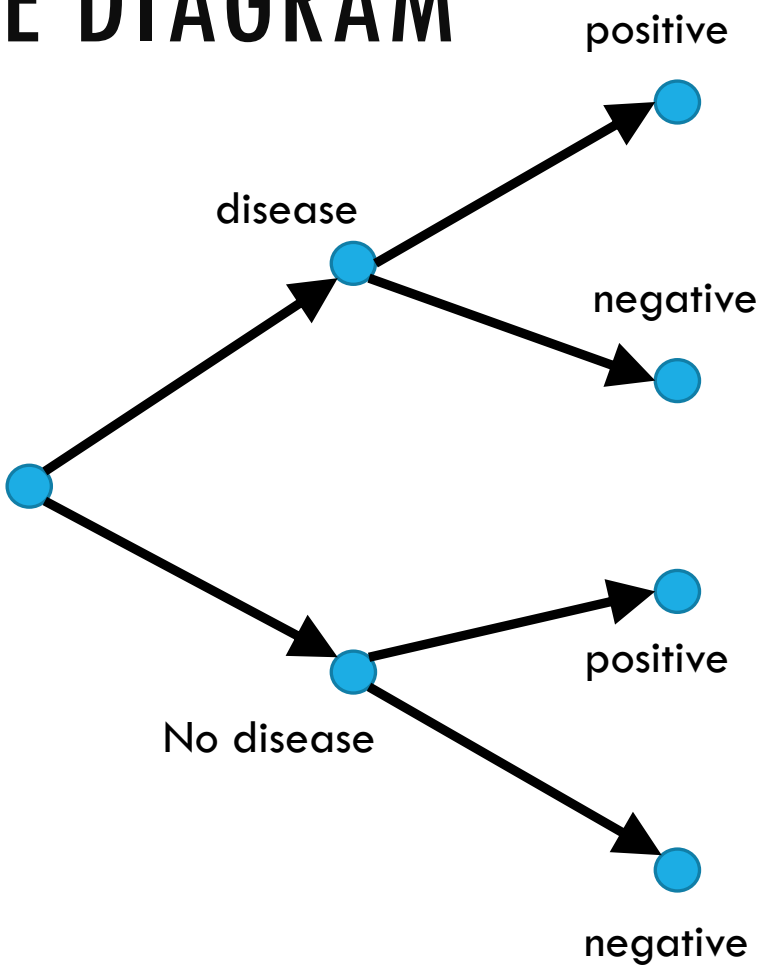
$\emptyset$

{(disease, positive),
(disease, negative)}

{(no disease, positive),
(no disease, negative)}

Ω

TREE DIAGRAM



Outcome Space Ω

(disease, positive)

(disease, negative)

(no disease, positive)

(no disease, negative)

Example Event Spaces:

Event Space E

$\emptyset$

{(disease, positive)}

{(disease, negative)}

{(no disease, positive)}

{(no disease, negative)}

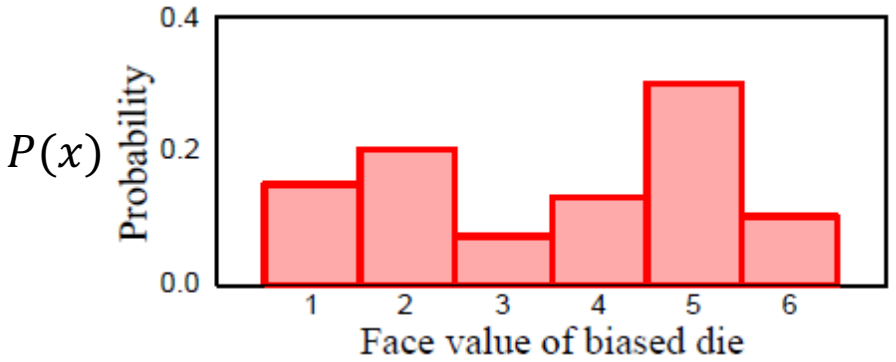
Ω

PROBABILITY DISTRIBUTIONS

A probability distribution P over (Ω, E) is a mapping from events in E to real values that satisfies the following conditions, i.e. axioms of probability:

1. Non-negativity, i.e. $P(\alpha) \geq 0, \ \forall \ \alpha \in E$.
2. Probability of all outcomes sums to 1, i.e. $P(\Omega) = 1$.
3. Mutually disjoint events: If $\alpha, \beta \in E$ and $\alpha \cap \beta = \emptyset$, then $P(\alpha \cup \beta) = P(\alpha) + P(\beta)$.

Discrete: Probability mass function, $P(x)$



$$Val(X) = \{1,2,3,4,5,6\}$$

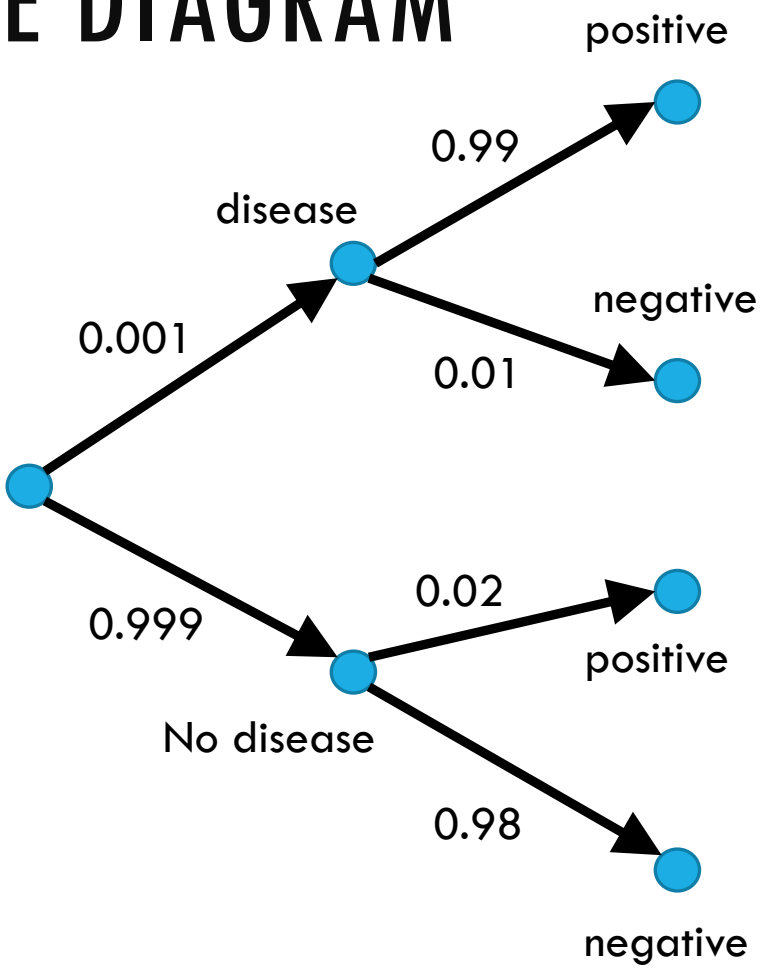
$$\sum_{i=1}^{K} P\left(X = x^i\right) = 1$$

$$0 \leq P\left(X = x^i\right) \leq 1, \ \forall\, i = 1, \ldots K$$

$$K = |Val(X)|$$

Images Source: "Computer Vision: Models, Learning, and Inference", Simon Prince

TREE DIAGRAM



Event Space E

$\emptyset$

{(disease, positive)}

0.001×0.99

{(disease, negative)}

0.001×0.01

{(no disease, positive)}

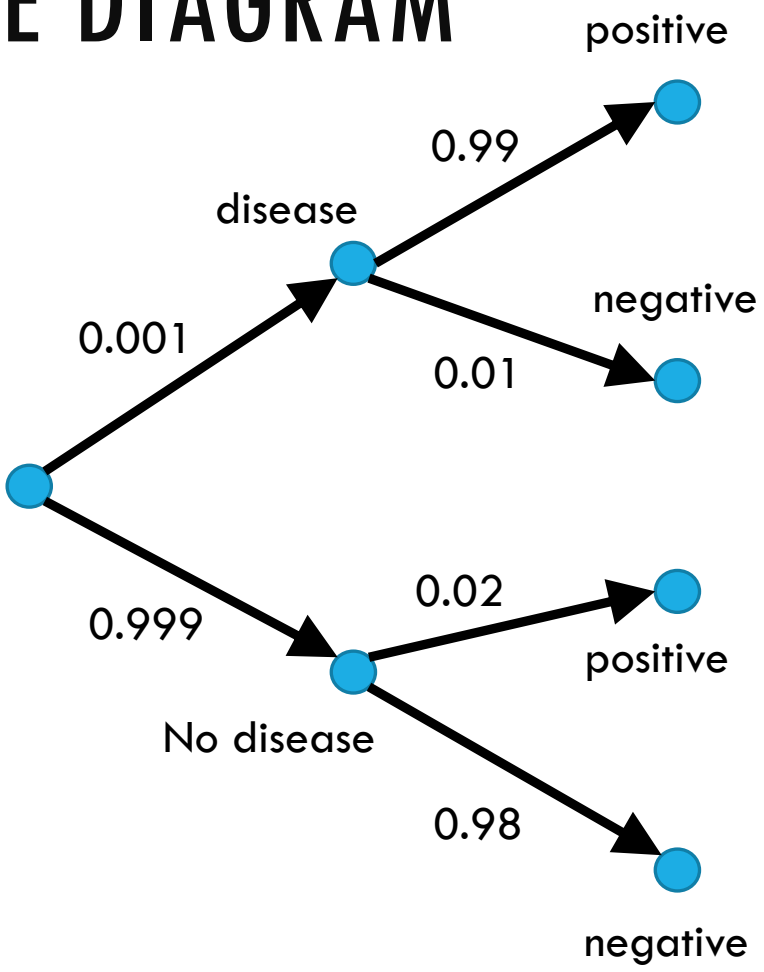
0.999×0.02

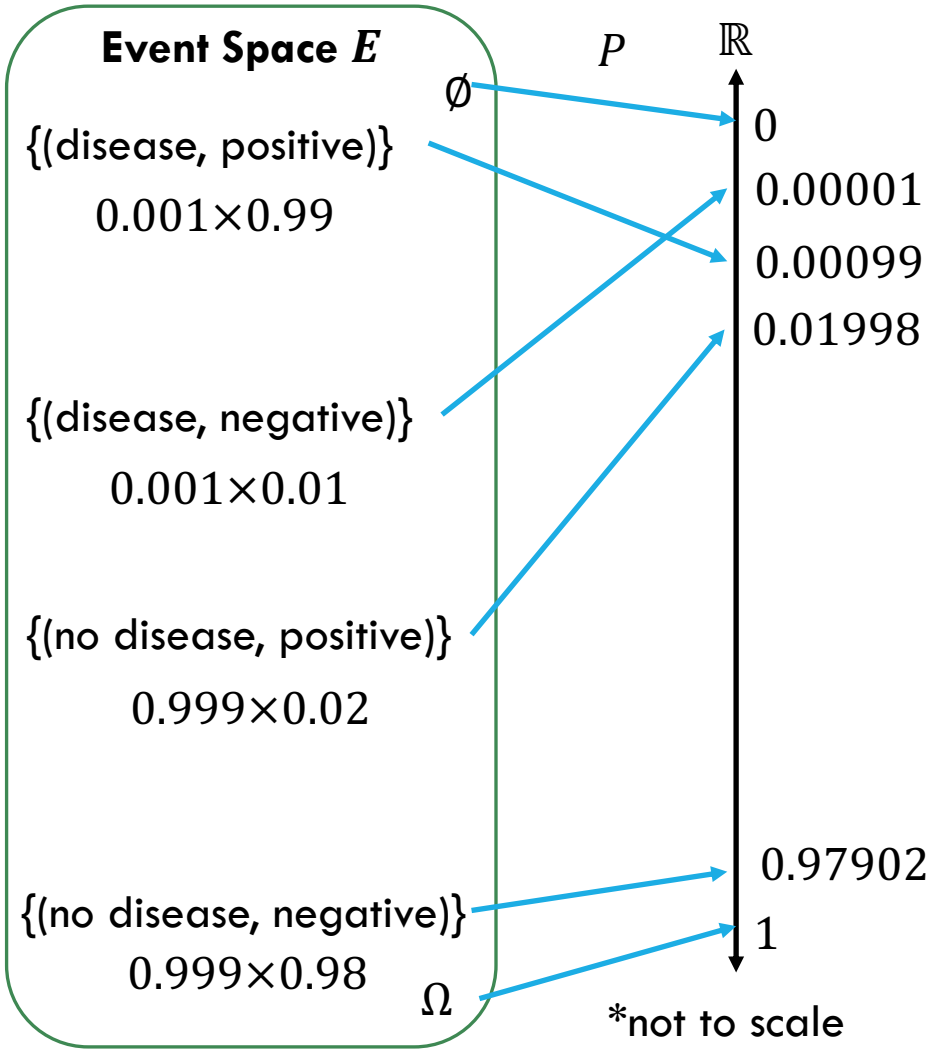
{(no disease, negative)}

0.999×0.98

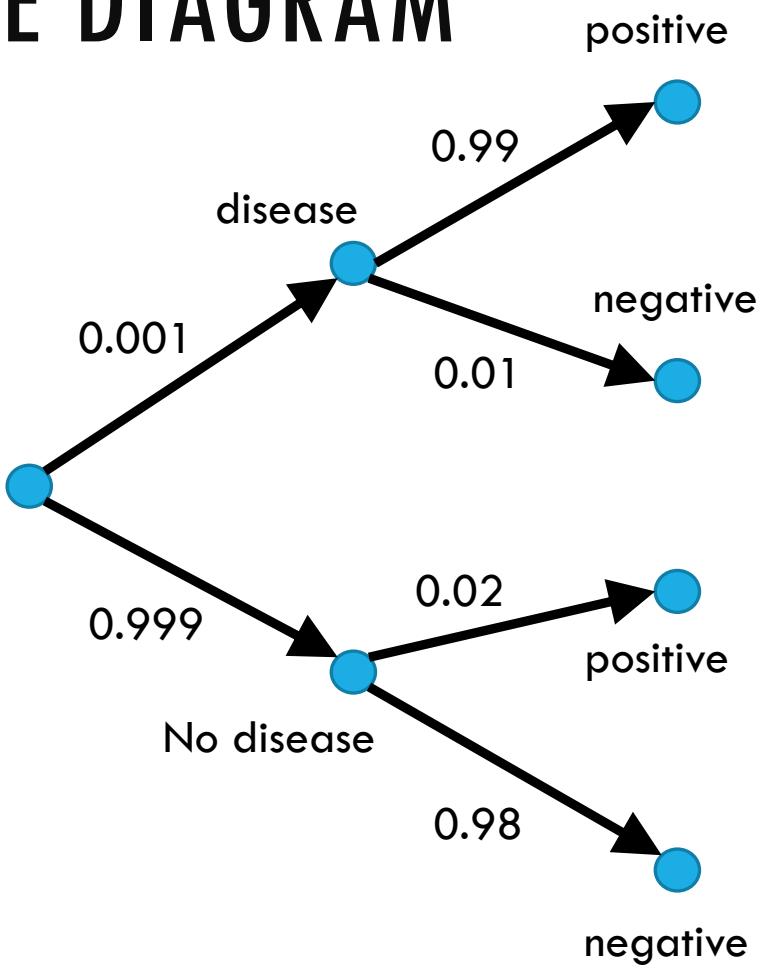
Ω

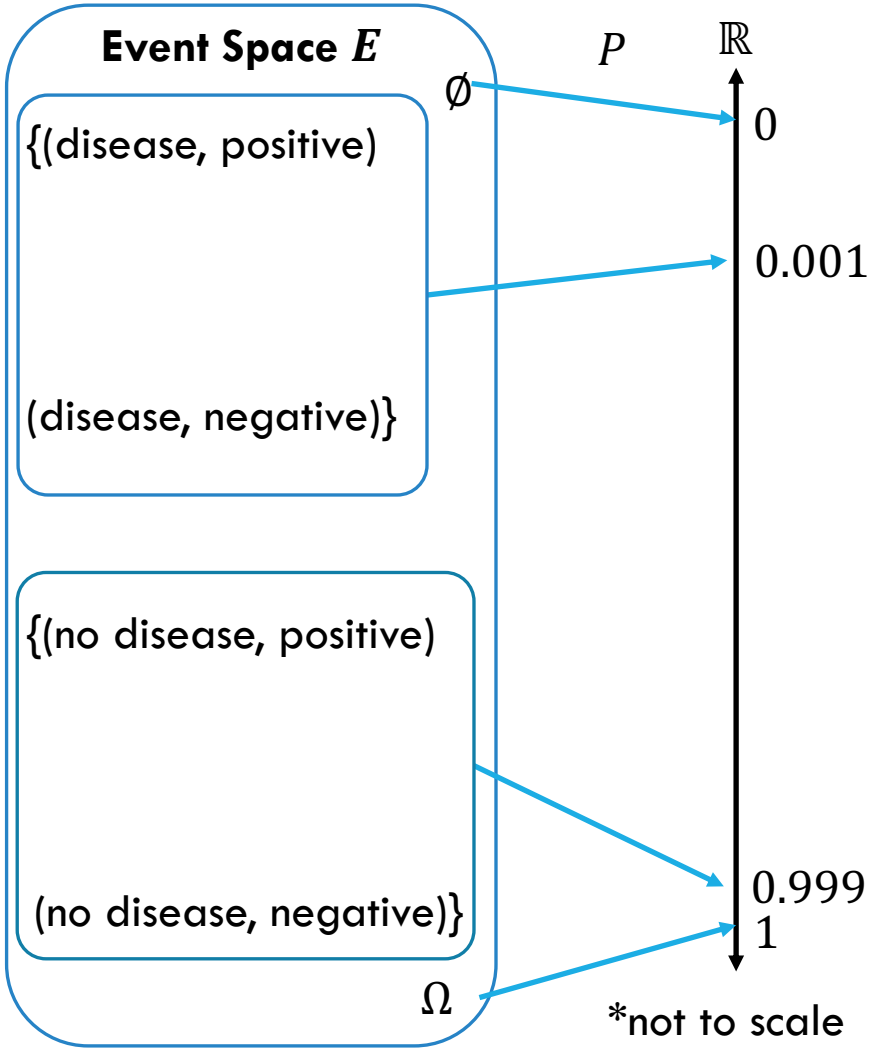
TREE DIAGRAM



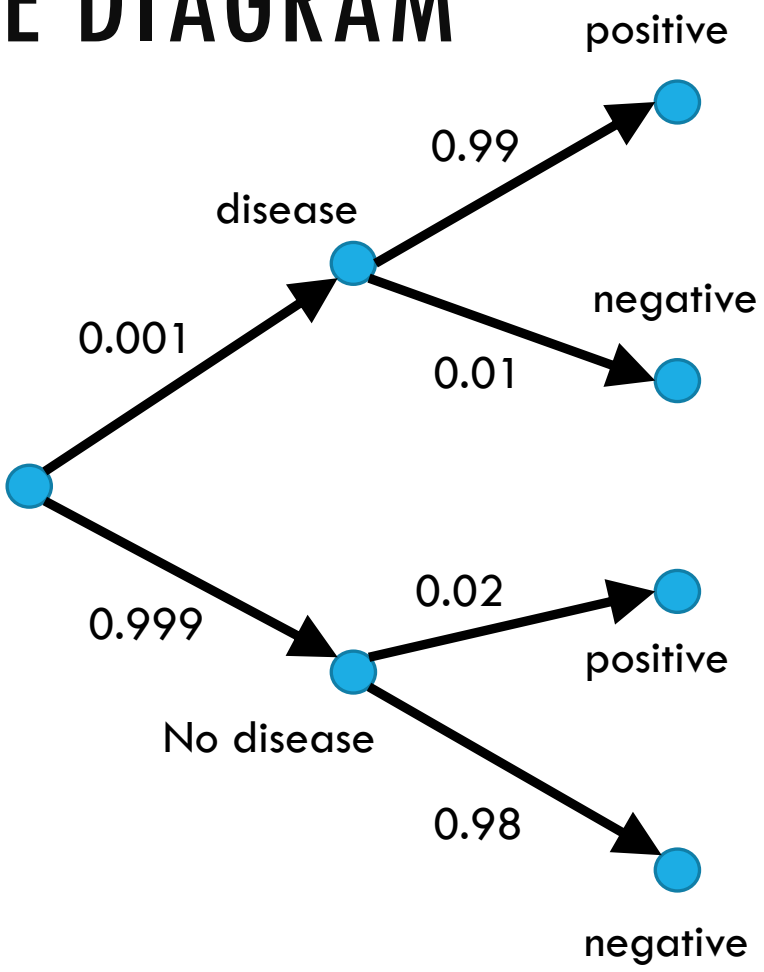


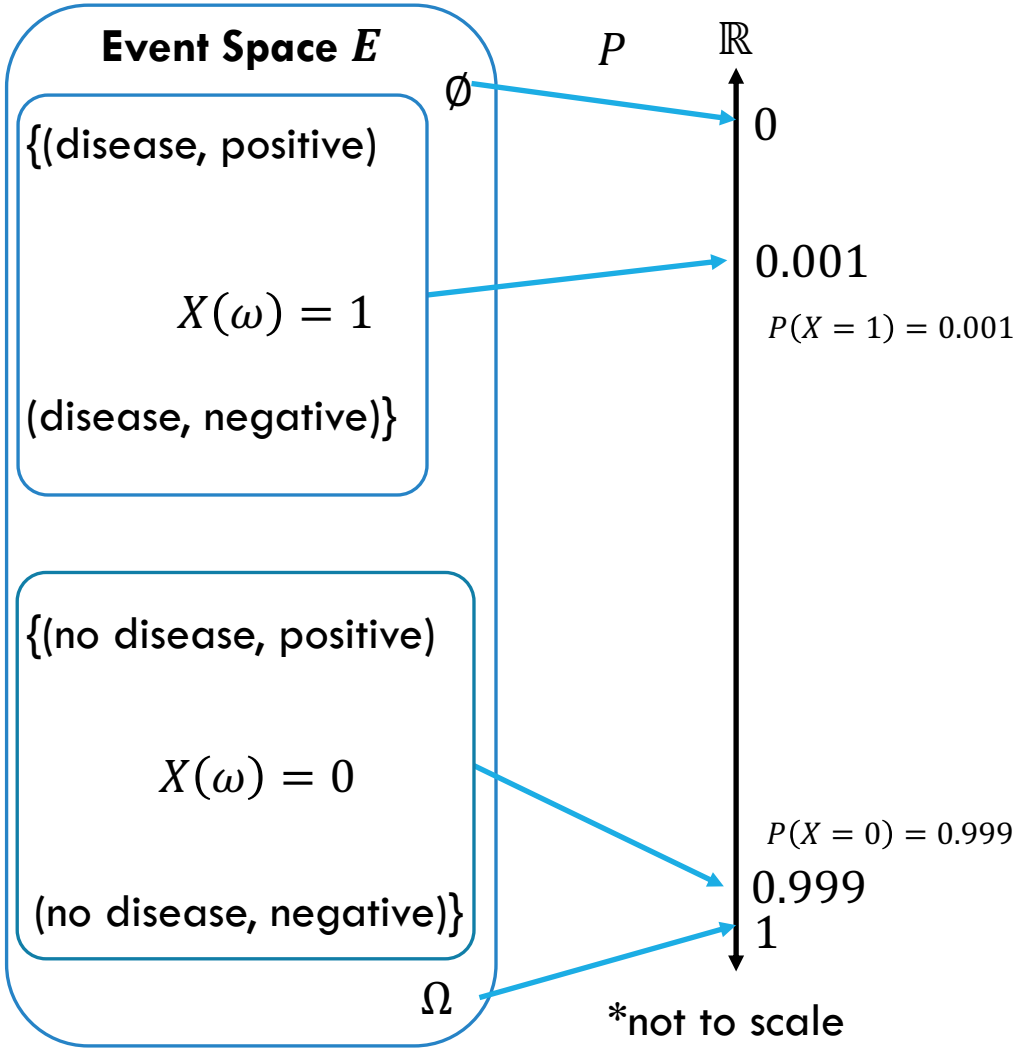
TREE DIAGRAM





TREE DIAGRAM





RANDOM VARIABLES

A random variable, denoted as X (upper case), is the machinery for discussing attributes and their values in different outcomes.

More formally, it is a function $X: \Omega \rightarrow S$ that maps a set of possible **outcomes** Ω to a space S

- S usually a subset of $\mathbb{R}$, but can be other sets.

RANDOM VARIABLES

The set of values that a random variable X can take is denoted as $Val(X)$.

A lower case letter, e.g. x, is used to refer to a generic value of a random variable X, a.k.a. realization of the random variable.

Example: We write $P(X = x) \geq 0$ for all $x \in Val(X)$.

$P(x)$ is often used as a shorthand notation for $P(X = x)$.

We use the notation x^i to represent a specific value of X.

INDICATOR RANDOM VARIABLES

Indicator random variable maps every **outcome** to either 0 or 1.

For example: whether you have the disease

$$\Omega = \{(d,\oplus), (\neg d,\oplus), (d,\ominus), (\neg d,\ominus)\}$$

$$X(\omega) = \begin{cases} 1 \text{ if } (d,\oplus) \\ 1 \text{ if } (d,\ominus) \\ 0 \text{ if } (\neg d,\oplus) \\ 0 \text{ if } (\neg d,\ominus) \end{cases}$$

$X = I_\alpha(\omega)$ where $\alpha \in E$ is the event where you have the disease

INDICATOR RANDOM VARIABLES

For example: 3 independent, unbiased coin tosses.

$$\Omega = \{ \text{HHH}, \text{TTT}, \text{HHT}, \text{HTH}, \text{HTT}, \text{THH}, \text{THT}, \text{TTH} \}$$

$$X(\omega) = \begin{cases} 1 \text{ if HHH or TTT} \\ 0 \text{ otherwise} \end{cases}$$

$X = I_\alpha(\omega)$ where $\alpha \in E$ is the event where all three coins match.

RANDOM VARIABLES: EXAMPLES

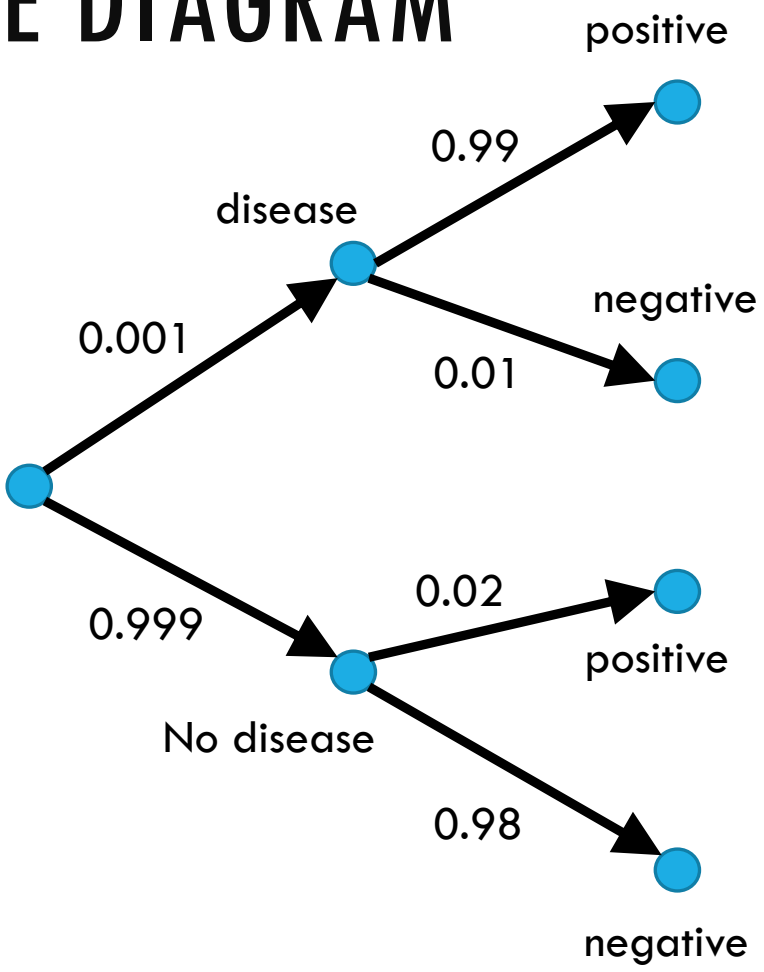
Random variables with discrete values

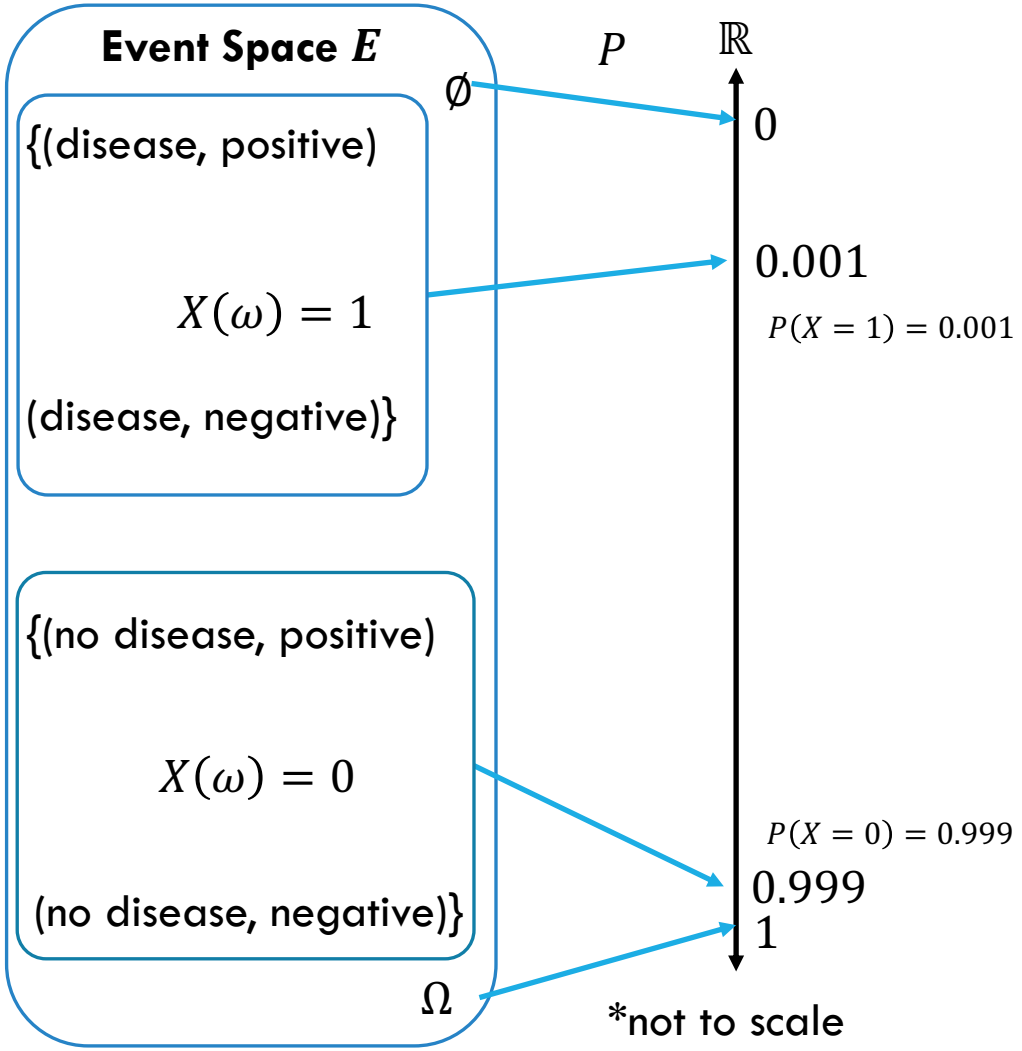
- Rolling a six-faced die: $Val(X) = \{1,2, \dots, 6\}$
- Weather conditions: $Val(X) = \{$"rain", "cloud", "snow", "sun", "wind"$\}$
- Number of people on the next train: $Val(X) = \mathbb{Z}_{\geq 0}$

Continuous random variables

- Time taken to finish an exam: $Val(X) = [1,2]$ hours
- Height of a tree: $Val(X) = \mathbb{R}_{>0}$
- Ambient Temperature: $Val(X) = \mathbb{R}$

TREE DIAGRAM



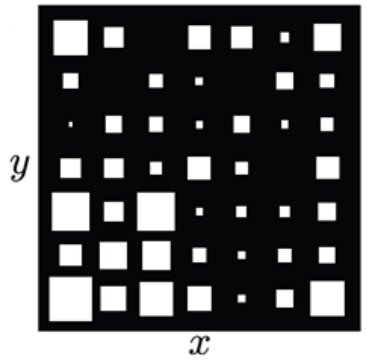


JOINT PROBABILITY

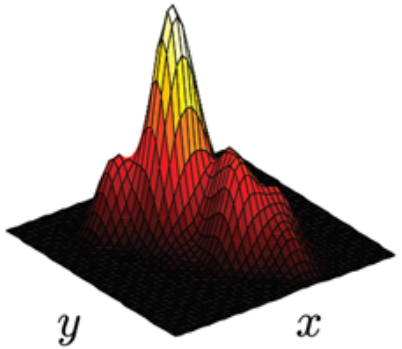
Consider all combination of events of two random variables X and Y.

Some combinations of outcomes are more likely than others.

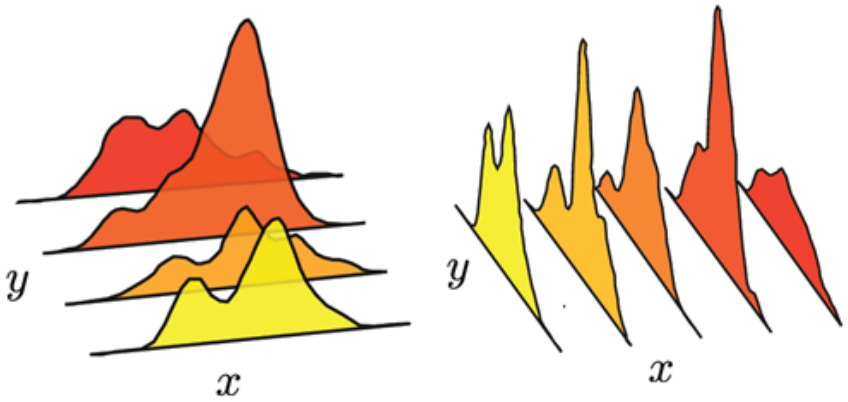
Discrete



Continuous



Discrete-Continuous



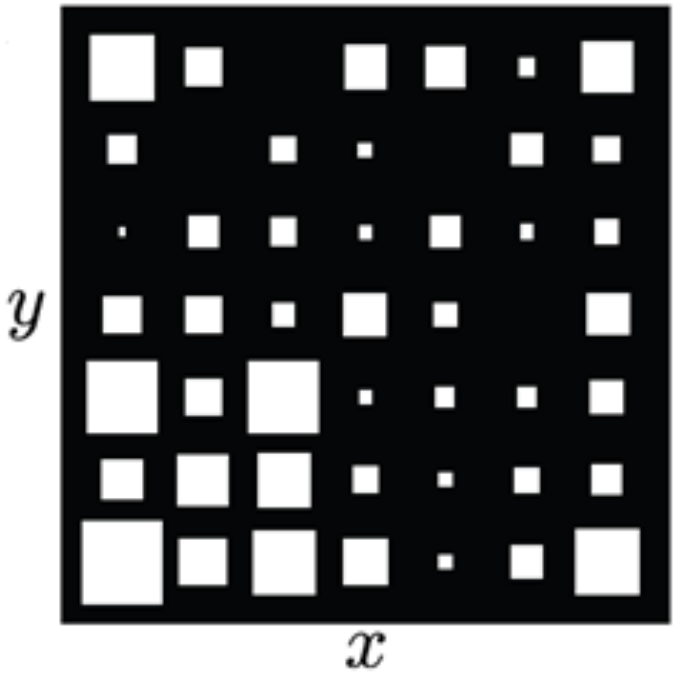
Images Source: "Computer Vision: Models, Learning, and Inference", Simon Prince

JOINT PROBABILITY

This is captured in the joint probability distribution $p(x, y)$.

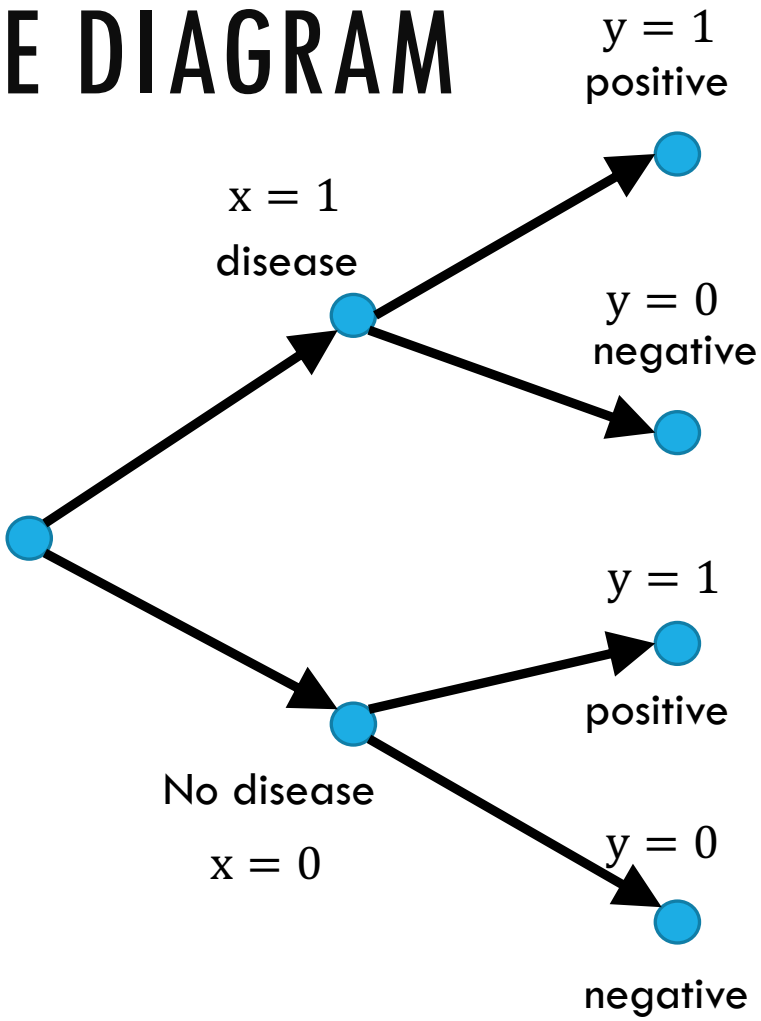
Read as "probability of X and Y".

Can be more than two random variables, i.e. $p(a, b, c, \dots)$.



Images Source: "Computer Vision: Models, Learning, and Inference", Simon Prince

TREE DIAGRAM



We can now have:

$$p(x, y) = P(X = x \text{ and } Y = y)$$

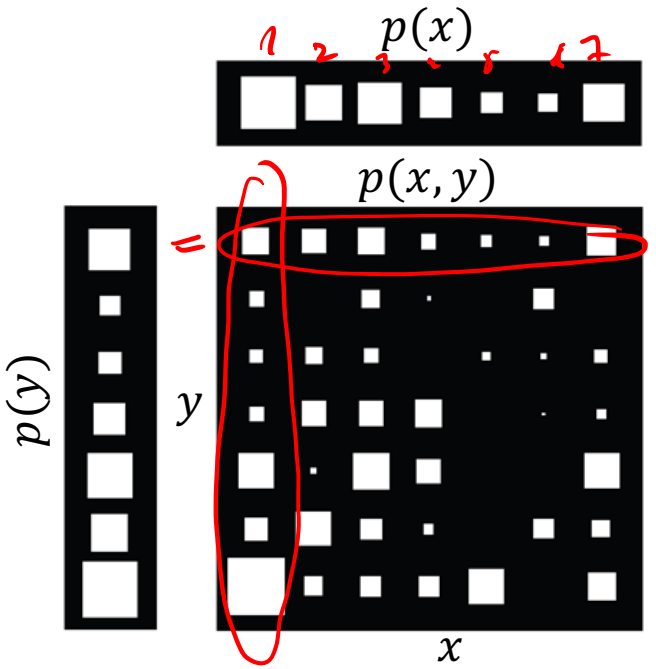
MARGINALIZATION

Recover probability distribution of any variable in a joint distribution by integrating (or summing) over all other variables.

Also known as the "sum rule" of probability.

$$p(x) = \sum_{y} p(x, y)$$

$$p(y) = \sum_{x} p(x, y)$$

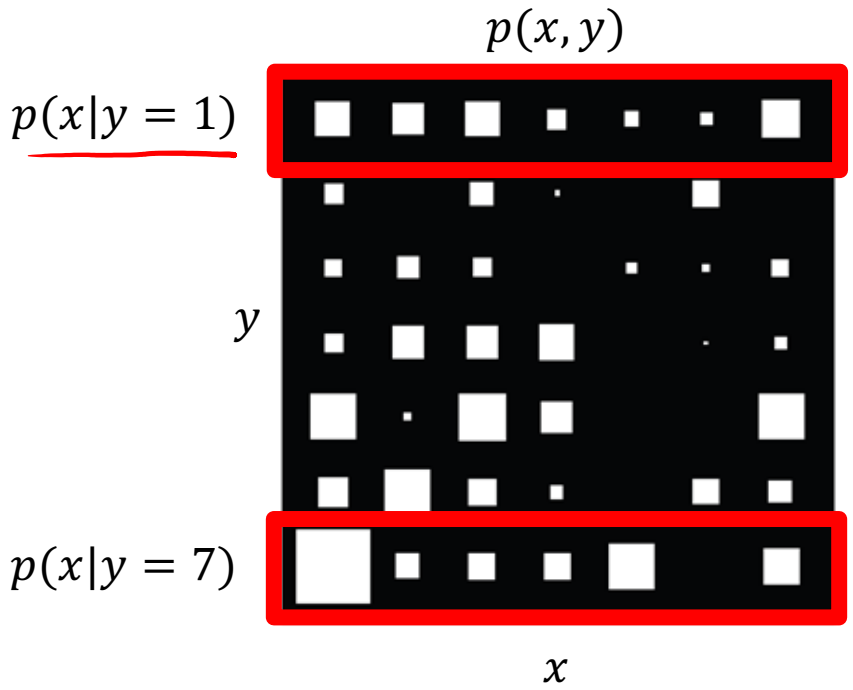


Images Source: "Computer Vision: Models, Learning, and Inference", Simon Prince

CONDITIONAL PROBABILITY

$p(x|Y = y^*)$: "probability of X given Y $= y^*$".

Relative propensity of the random variable X to take different outcomes given that the random variable Y is fixed to value y^*.



Images Source: "Computer Vision: Models, Learning, and Inference", Simon Prince

PROBABILITY: CONDITIONAL PROBABILITY

$$P(x|Y=y^*) = \frac{p(x,Y=y^*)}{\sum_x p(x,Y=y^*)} = \frac{p(x,Y=y^*)}{p(Y=y^*)}$$

Usually written in compact form:

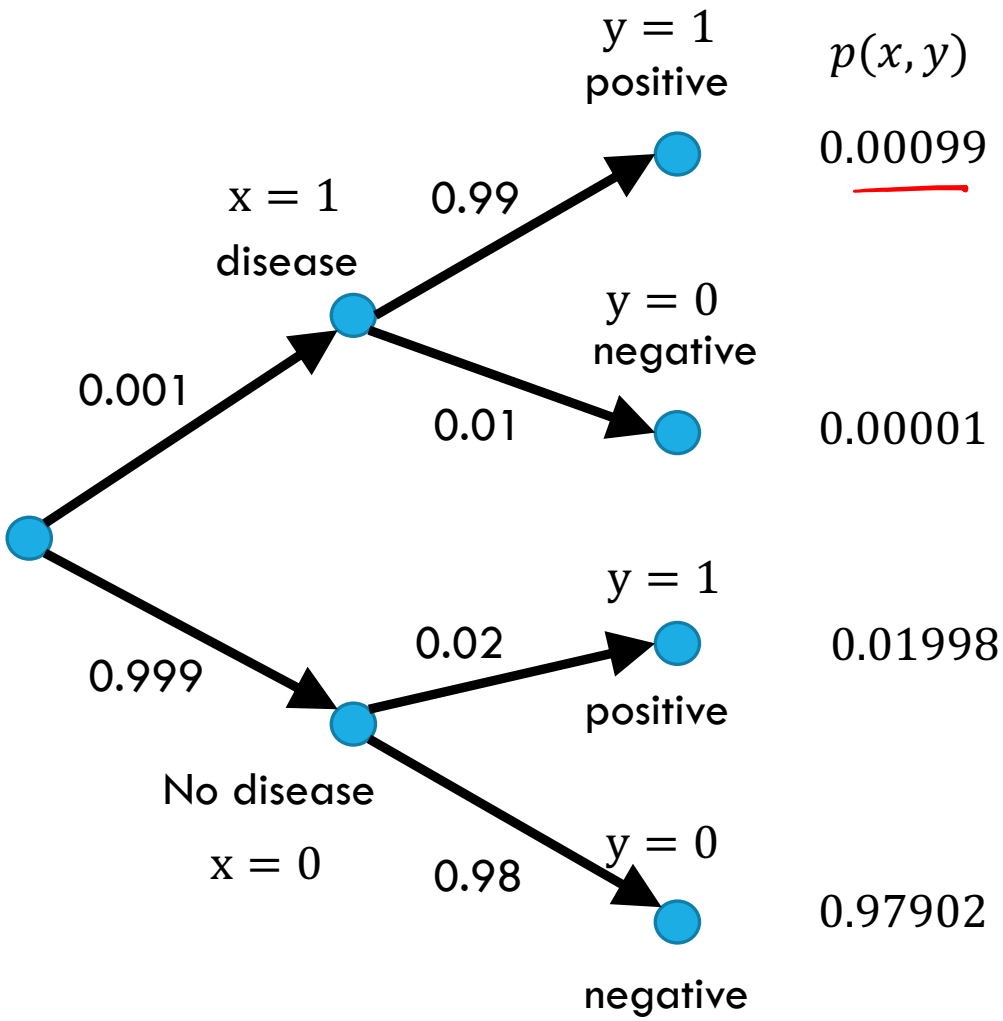
$$p(x|y) = \frac{p(x,y)}{p(y)}$$

Which can be re-arranged to give:

$$p(x,y) = p(x|y)p(y)$$
$$p(x,y) = p(y|x)p(x)$$

known as "chain rule" or "product rule" of probability.

WHAT IS THE PROBABILITY I HAVE THE DISEASE GIVEN THE TEST IS POSITIVE?



$p(x, y)$

0.00099

0.00001

0.01998

0.97902

We want: $p(X = 1 \mid Y = 1)$

$$p(x|Y = 1) = \frac{p(x, Y = 1)}{p(Y = 1)} = \frac{p(x, Y = 1)}{\sum_x p(x, Y = 1)}$$

sum rule

$$p(Y = 1) = 0.00099 + 0.01998 = 0.02097$$

$$p(X = 1|Y = 1) = 0.00099/0.02097 = 0.0472 < 5\%$$

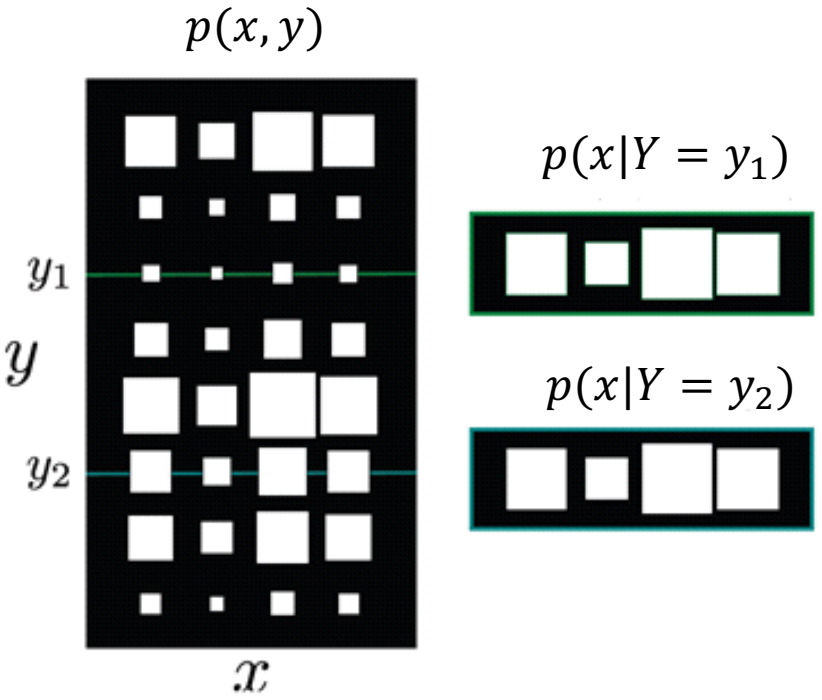
INDEPENDENCE

The independence of X and Y means that every conditional distribution is the same.

The value of Y tells us nothing about X and vice-versa.

$$p(x|y) = p(x)$$

$$p(y|x) = p(y)$$

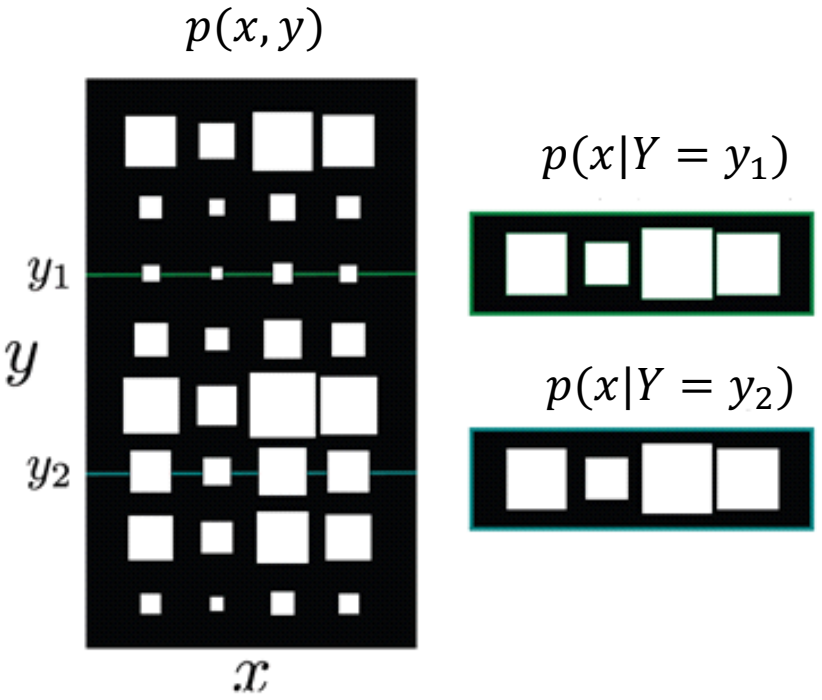


Images Source: "Computer Vision: Models, Learning, and Inference", Simon Prince

INDEPENDENCE

When variables are independent, the joint factorizes into a product of the marginals:

$$p(x,y) = p(x|y)p(y)$$
$$= p(x)p(y)$$



Images Source: "Computer Vision: Models, Learning, and Inference", Simon Prince

SUMMARY: SUM AND PRODUCT RULES

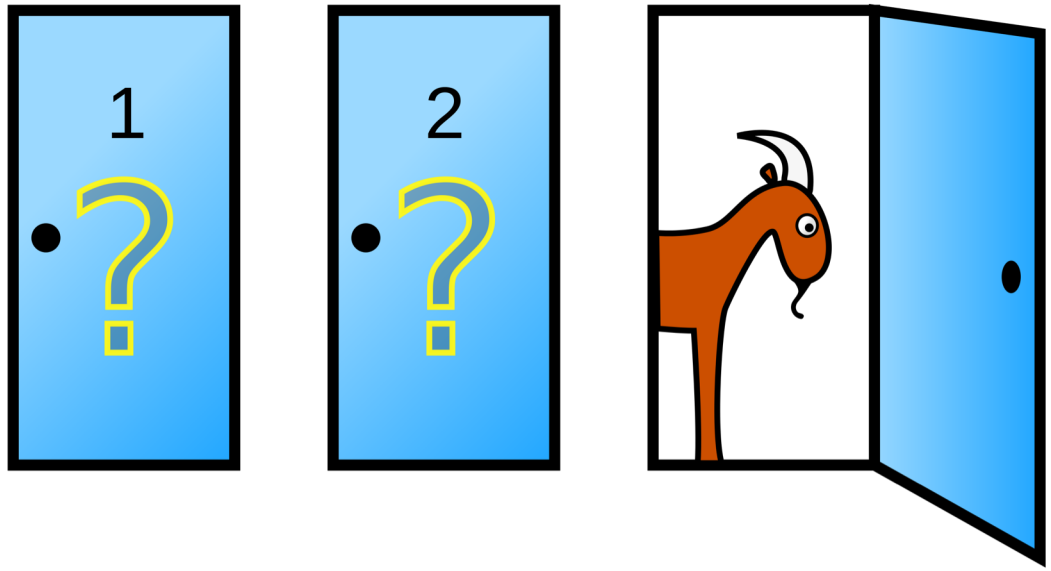
Sum rule:

$$p(x) = \sum_{y} p(x, y)$$

Product/Chain rule:

$$p(x, y) = p(x|y)p(y)$$

YOU'VE SEEN THIS PROBLEM BEFORE, RIGHT?



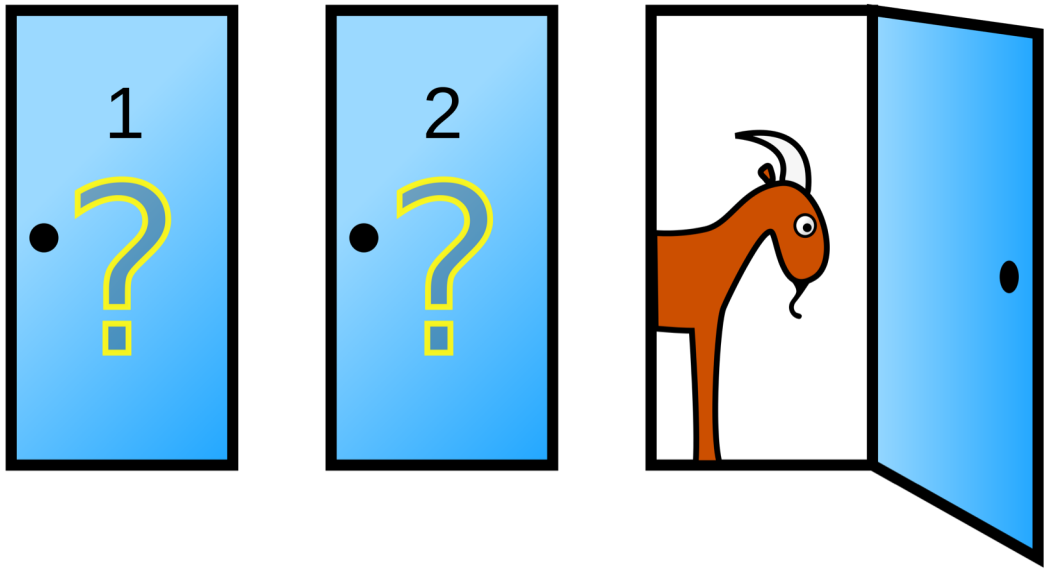
Should you switch?

A. Yes

B. No

You're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say Door 1, and the host, who knows what's behind the doors, opens another door, say Door 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice?

YOU'VE SEEN THIS PROBLEM BEFORE, RIGHT?



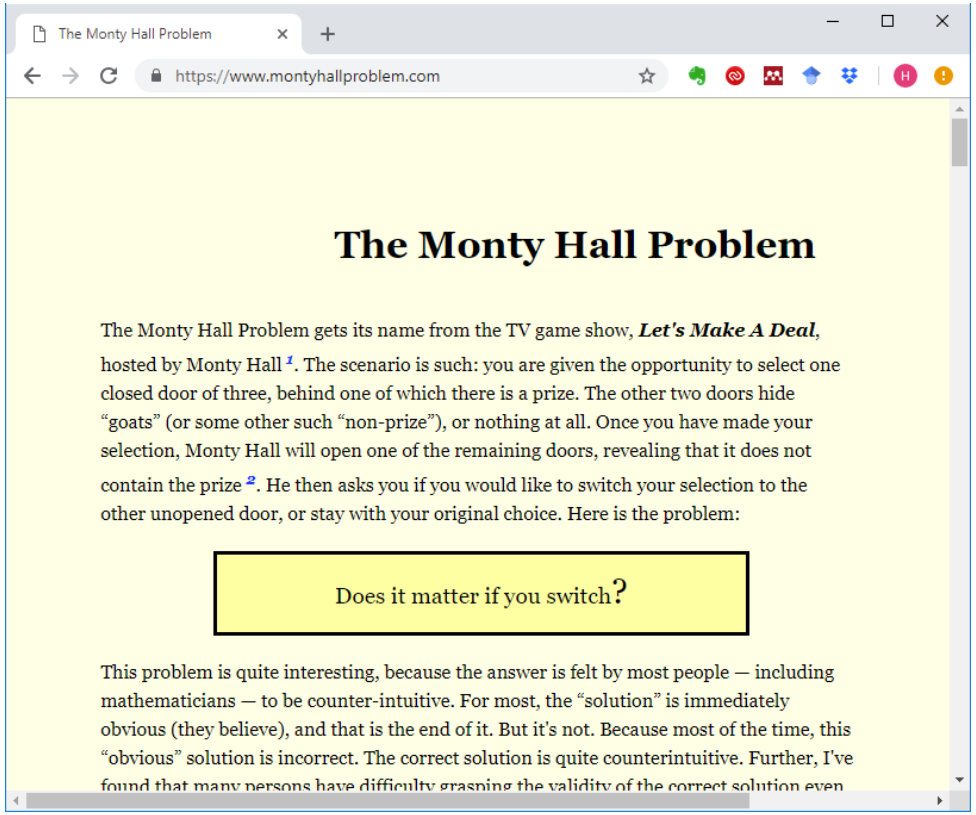
Should you switch?

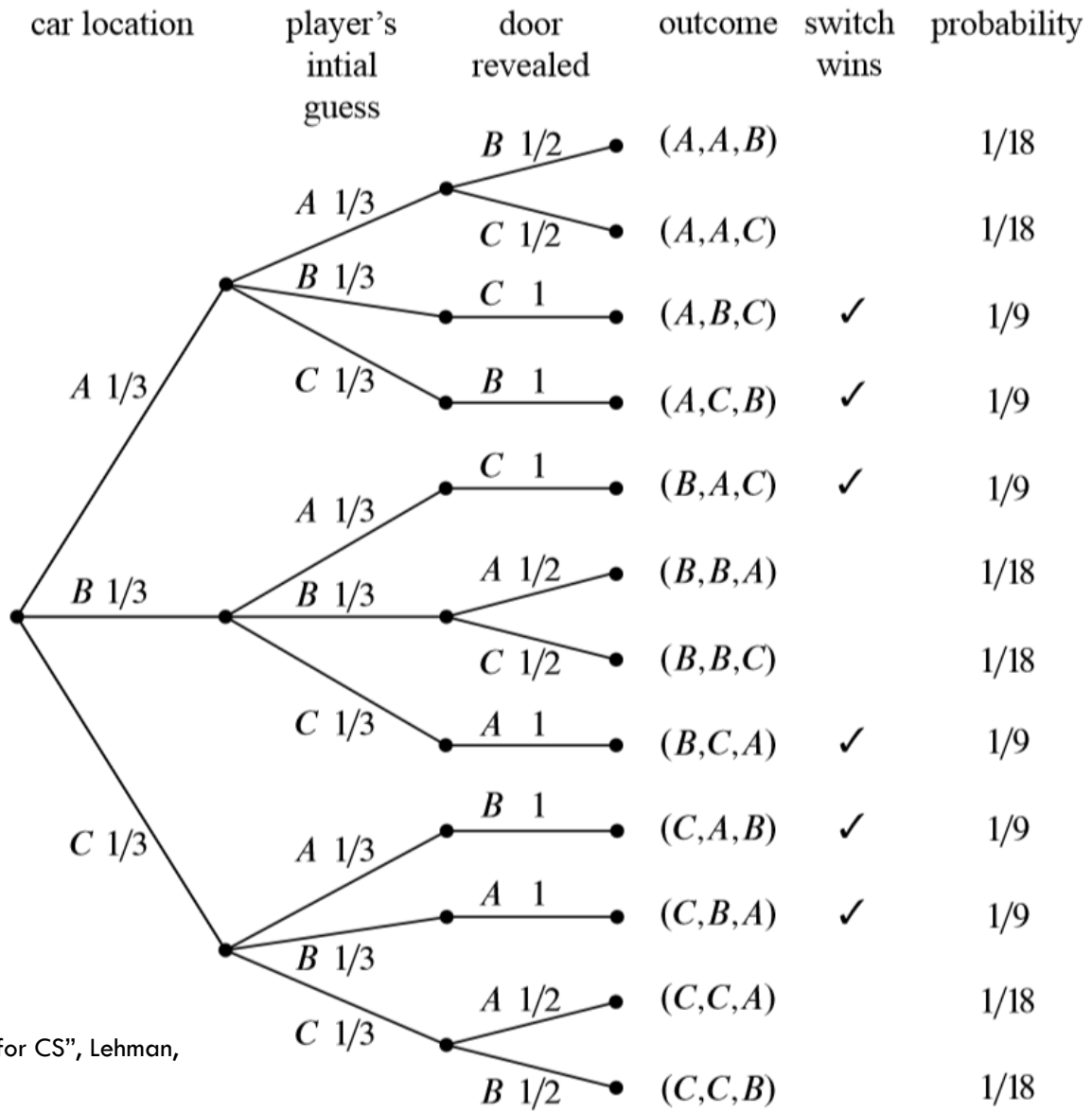
A. Yes

B. No

You're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say Door 1, and the host, who knows what's behind the doors, opens another door, say Door 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice?

MONTY-HALL PROBLEM





Adapted from "Math for CS", Lehman, Leighton and Meyer

$p(\text{switch wins}) = 6 \times \frac{1}{9} = \frac{2}{3}$

Should switch!

2 NUMBERS GAME

Team 1:

- Pick 2 **different** numbers between 0 and 10.
- Write each number on a piece of paper each.
- Turn the papers face down.

Team 2:

- Objective is to pick the **larger number**.
- Pick one of the pieces of paper.
- Have a peek at the number.
- **Decide:** do you keep this number or *switch*?

Question: Can Team 2 win more than 50% of the time?

TEAM 2 STRATEGY

Guess a number $Z \in [0, 10)$

Take a peek at one of the numbers, lets call this x

If $x \leq Z$, switch, else stick with x

STRATEGY ANALYSIS

Let $l < h$ be the numbers chosen by Team 1.

3 possible cases:

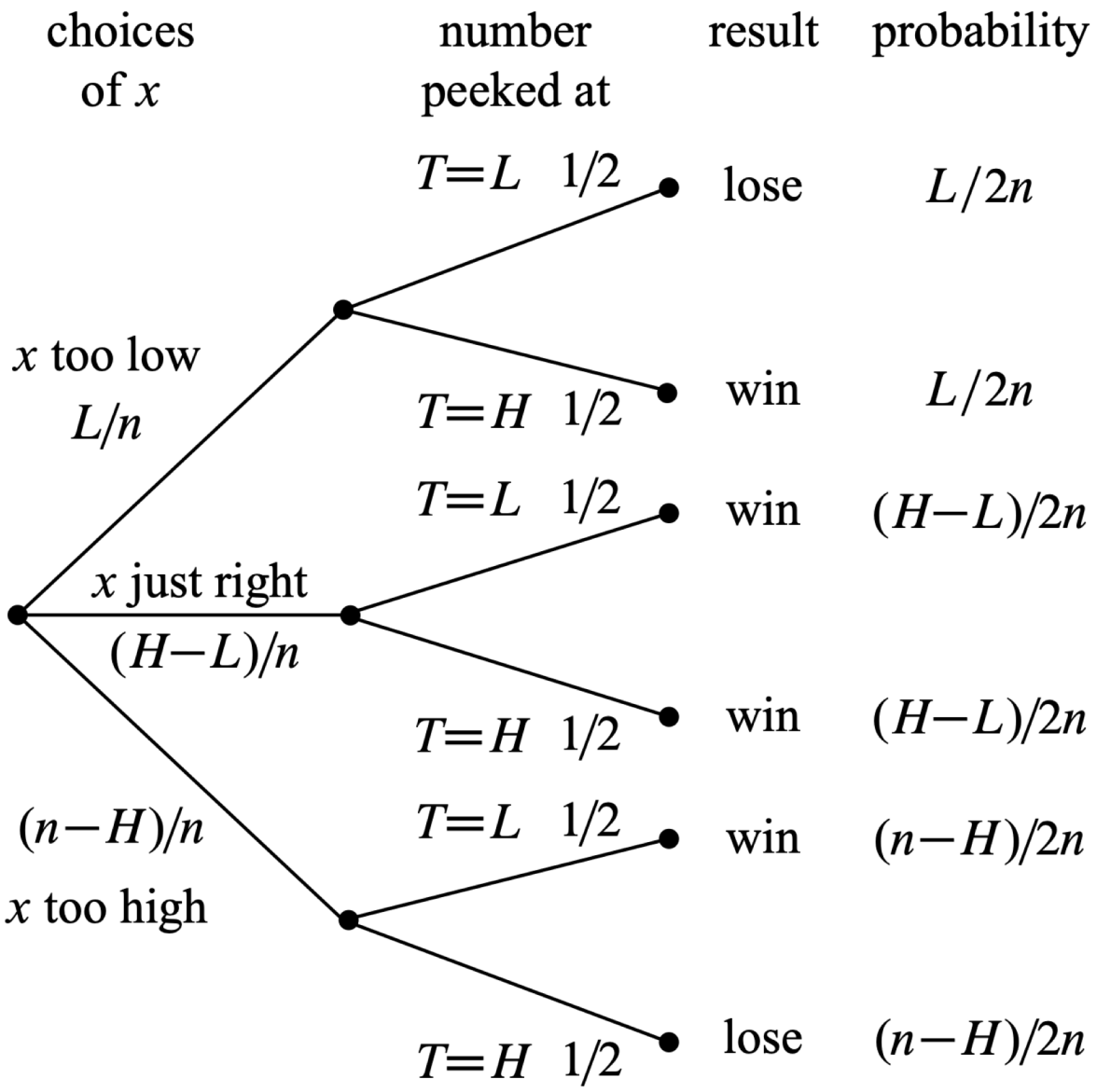
- **Case** M: $l \leq Z < h$:
  - Team 2 wins always!
  - $p(\text{win}|M) = 1$ and $p(M) \geq \frac{1}{10}$
- **Case** H: $h \leq Z$:
  - Team 2 switch. Only wins if picked l
  - $p(\text{win}|H) = \frac{1}{2}$
- **Case** L: $Z < l$:
  - Team 2 switch. Only wins if picked h
  - $p(\text{win}|L) = \frac{1}{2}$

$$p(win) \geq \left(1 \times \frac{1}{10}\right) + \frac{1}{2}\left(1 - \frac{1}{10}\right)$$

$$p(win) \geq \frac{11}{20}$$

More complete derivation in:
MIT Math for CS, Chapter 18.3.3
https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-042j-mathematics-for-computer-science-spring-2015/readings/MIT6_042JS15_Session31.pdf

IN GENERAL



More complete derivation in: MIT Math for CS, Chapter 18.3.3

https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-042j-mathematics-for-computer-science-spring-2015/readings/MIT6_042JS15_Session31.pdf

QUESTIONS?

Poll Everywhere

https://bit.ly/2LvG9bq

THE NEXT 2 DAYS

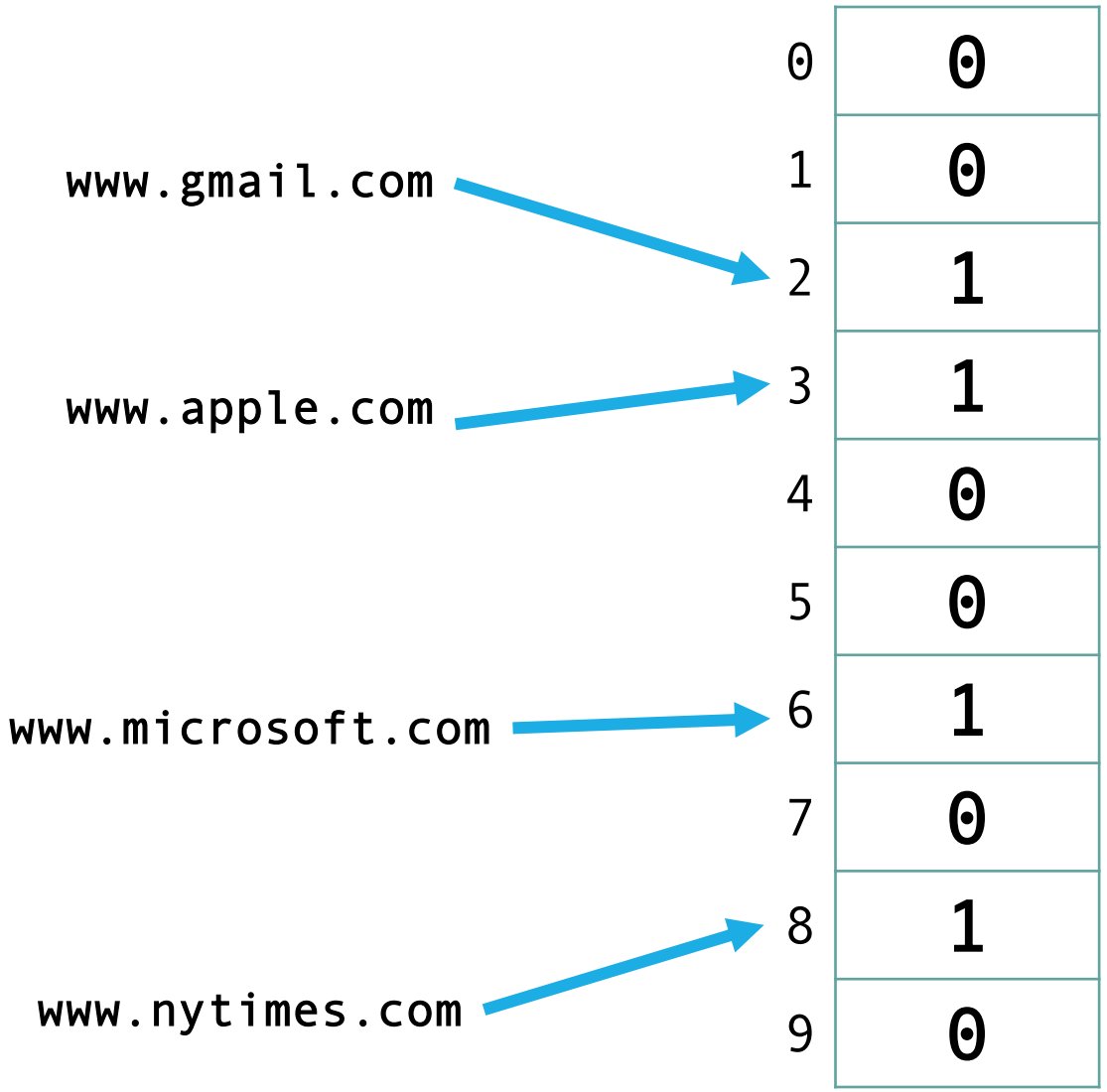
- Probability Review
- Bloom Filters
- Cuckoo Hashing

IDEA: STORE LESS!

Only set k bits per item in a table of size m.

To start, let k = 1

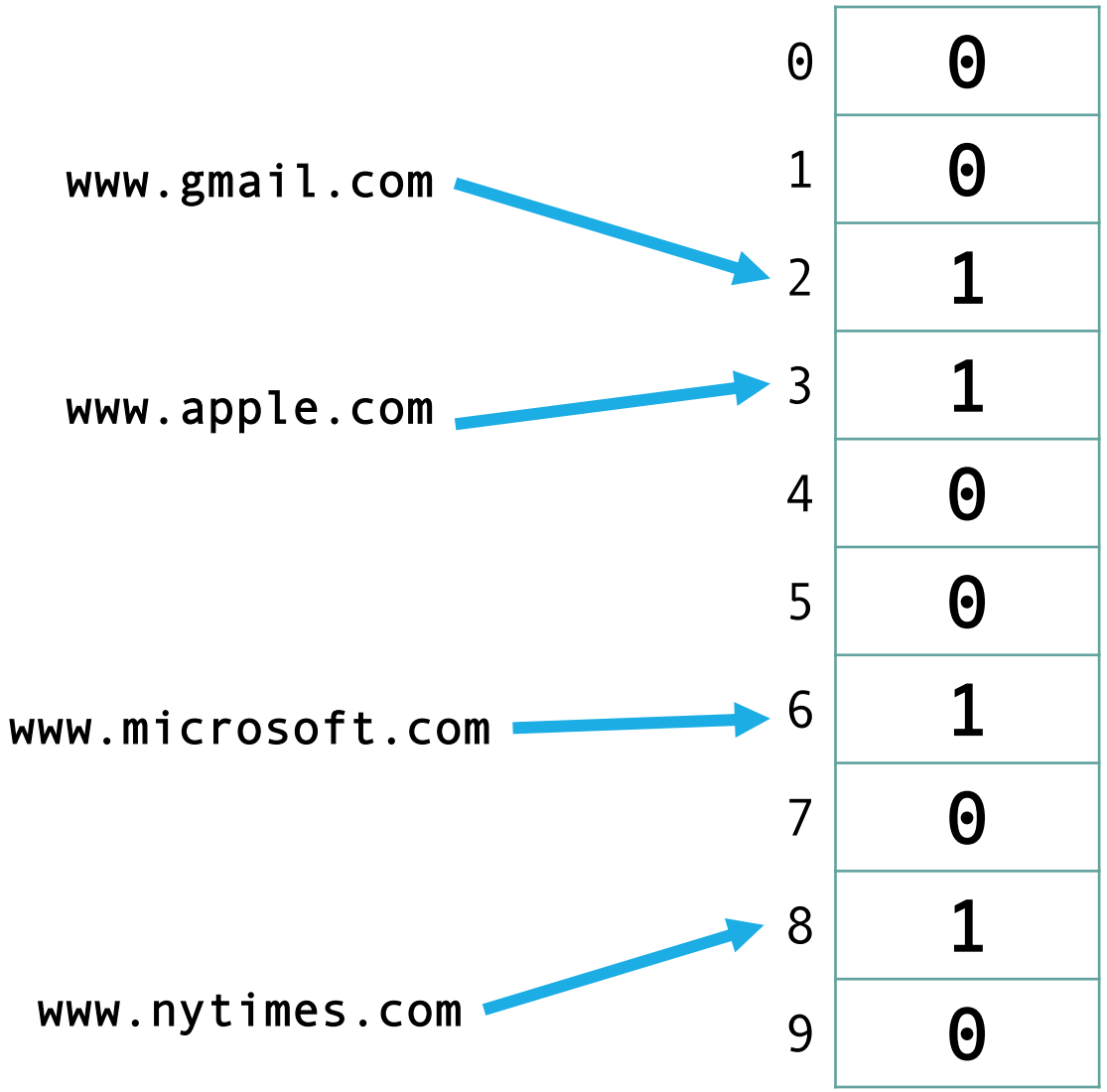
- For each item, we only set 1 bit!



IDEA: STORE LESS!

```
insert(key)
    h = hash(key)
    m_table[h] = 1

lookup(key)
    h = hash(key)
    return (m_table[h] == 1)
```



PROBABILITY OF A FALSE POSITIVE?

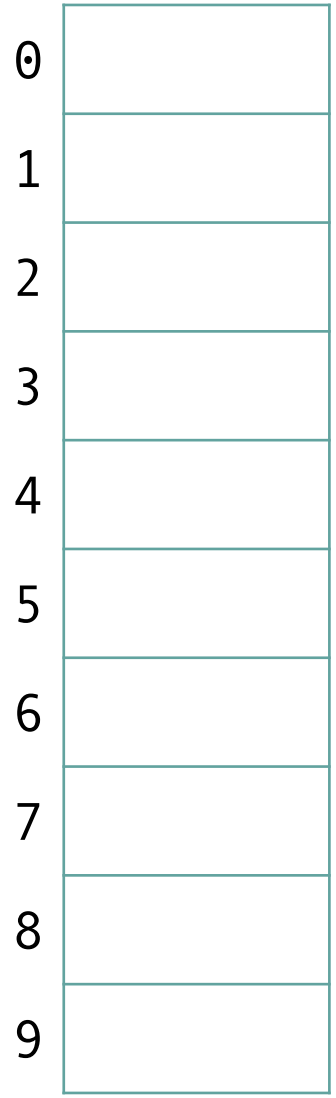
Table size is m, we store n elements

Set $k = 1$ bit per item

Test an item **not** in the set.

$$p(\text{false positive}) = 1 - p(\text{hits a cell with } 0)$$

$$p(\text{hits cell with } 0) = \underbrace{\left(1 - \frac{1}{m}\right)^n}_{\text{Elements are independently hashed}} \approx e^{-\frac{n}{m}}$$



PROBABILITY OF A FALSE POSITIVE?

Test an item **not** in the set.

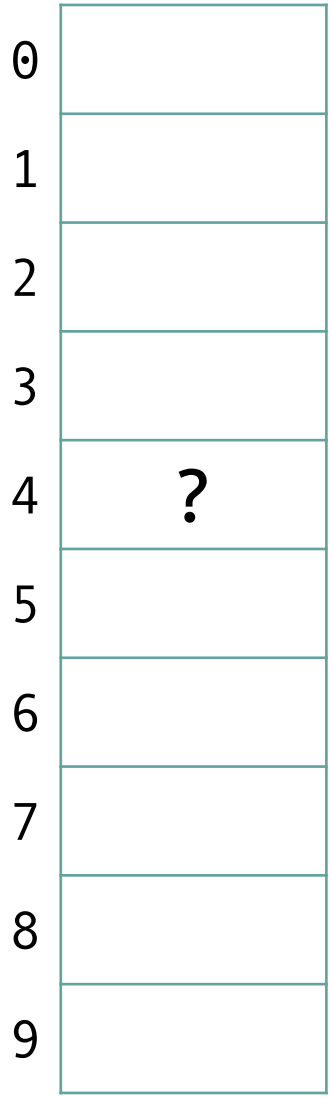
Let's say the item hits cell 4.

What is the probability that the cell is a 0?

- No previous insertion has set it to 1

Say $n = 1$ (only 1 item inserted in the table)

- Then $p(\text{cell is } 0) = \left(1 - \frac{1}{m}\right)$ (Simple uniform hashing assumption)



PROBABILITY OF A FALSE POSITIVE?

Say $n = 1$ (only 1 item inserted in the table)

- Then $p(\text{cell is } 0) = \left(1 - \frac{1}{m}\right)$ (Simple uniform hashing assumption)

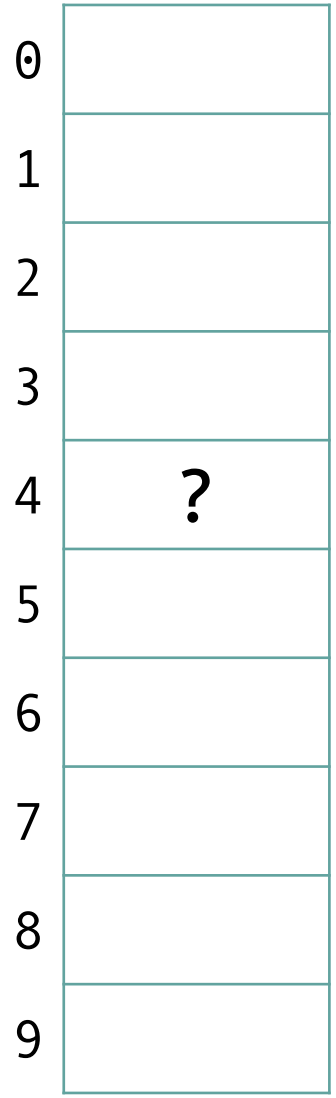
Say n = 2 (2 items inserted)

- Then $p(\text{cell is } 0) = \left(1 - \frac{1}{m}\right)^2$ (Simple uniform hashing assumption)

…

For n items inserted

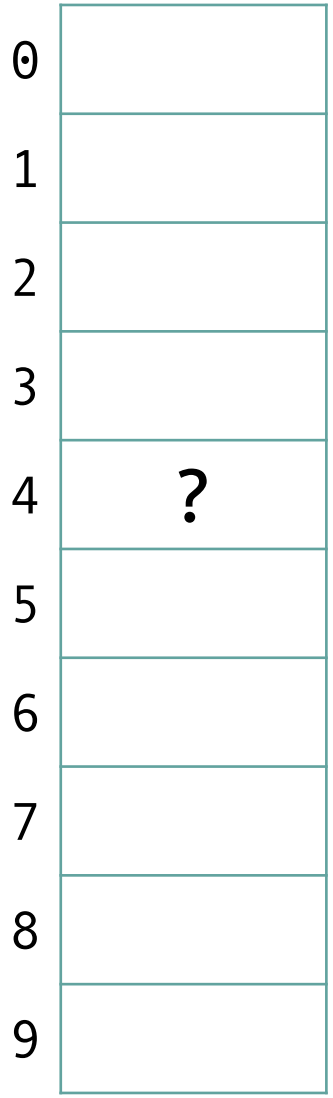
- Then $p(\text{cell is } 0) = \left(1 - \frac{1}{m}\right)^n$ (Simple uniform hashing assumption)



PROBABILITY OF A FALSE POSITIVE?

For n items inserted

- Then $p(\text{cell is } 0) = \left(1 - \frac{1}{m}\right)^n \approx e^{-\frac{n}{m}}$
- Note:
  - $\lim_{n\to\infty} \left(1 - \frac{1}{m}\right)^n = e^{-\frac{n}{m}}$



PROBABILITY OF A FALSE POSITIVE?

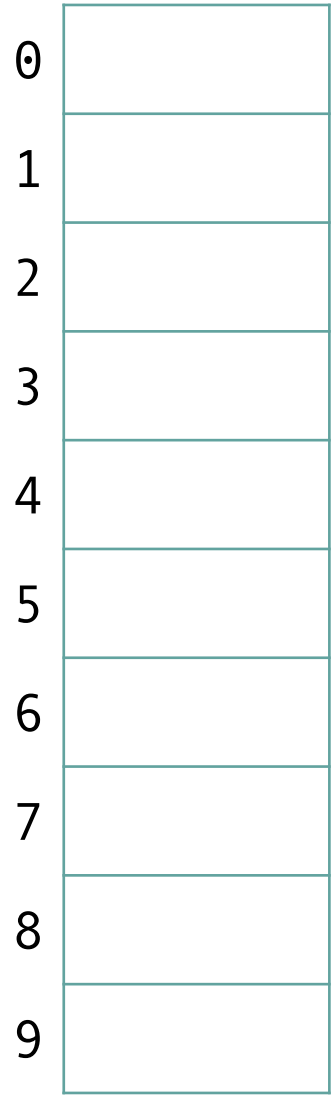
Table size is m, we store n elements

Set $k = 1$ bit per item

Test an item **not** in the set.

$$p(\text{false positive}) = 1 - p(\text{hits a cell with } 0)$$

$$p(\text{hits cell with } 0) = \underbrace{\left(1 - \frac{1}{m}\right)^n}_{\text{Elements are independently hashed}} \approx e^{-\frac{n}{m}}$$



PROBABILITY OF A FALSE POSITIVE?

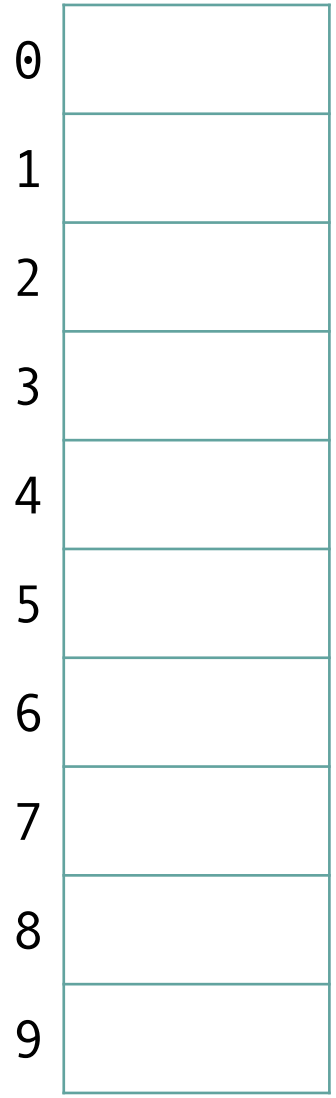
Table size is m, we store n elements

Set $k = 1$ bit per item

Test an item **not** in the set.

$$p(\text{false positive}) = 1 - p(\text{hits a cell with } 0)$$

$$p(\text{false positive}) \approx 1 - e^{-\frac{n}{m}}$$



LET'S SET P(FALSE POSITIVE)!

Given n items, say we want $\hat{p} = 10\% = 0.1$

Then,

- $p(\text{false positive}) = 1 - e^{-\frac{n}{m}} \leq \hat{p}$
- $\frac{n}{m} \leq \ln\left(\frac{1}{1-\hat{p}}\right)$
- $m \geq \frac{n}{\ln\left(\frac{1}{1-\hat{p}}\right)}$
- So, $m \geq 9.49n$

If you wanted the derivation:

$$1 - e^{-\frac{n}{m}} \leq \hat{p}$$
$$e^{-\frac{n}{m}} \geq 1 - \hat{p}$$
$$-\frac{n}{m} \geq \ln(1 - \hat{p})$$
$$\frac{n}{m} \leq \ln\left(\frac{1}{1-\hat{p}}\right)$$
$$m \geq \frac{n}{\ln\left(\frac{1}{1-\hat{p}}\right)}$$

LET'S SET P(FALSE POSITIVE)!

Given n items, say we want $\hat{p} = 1\% = 0.01$

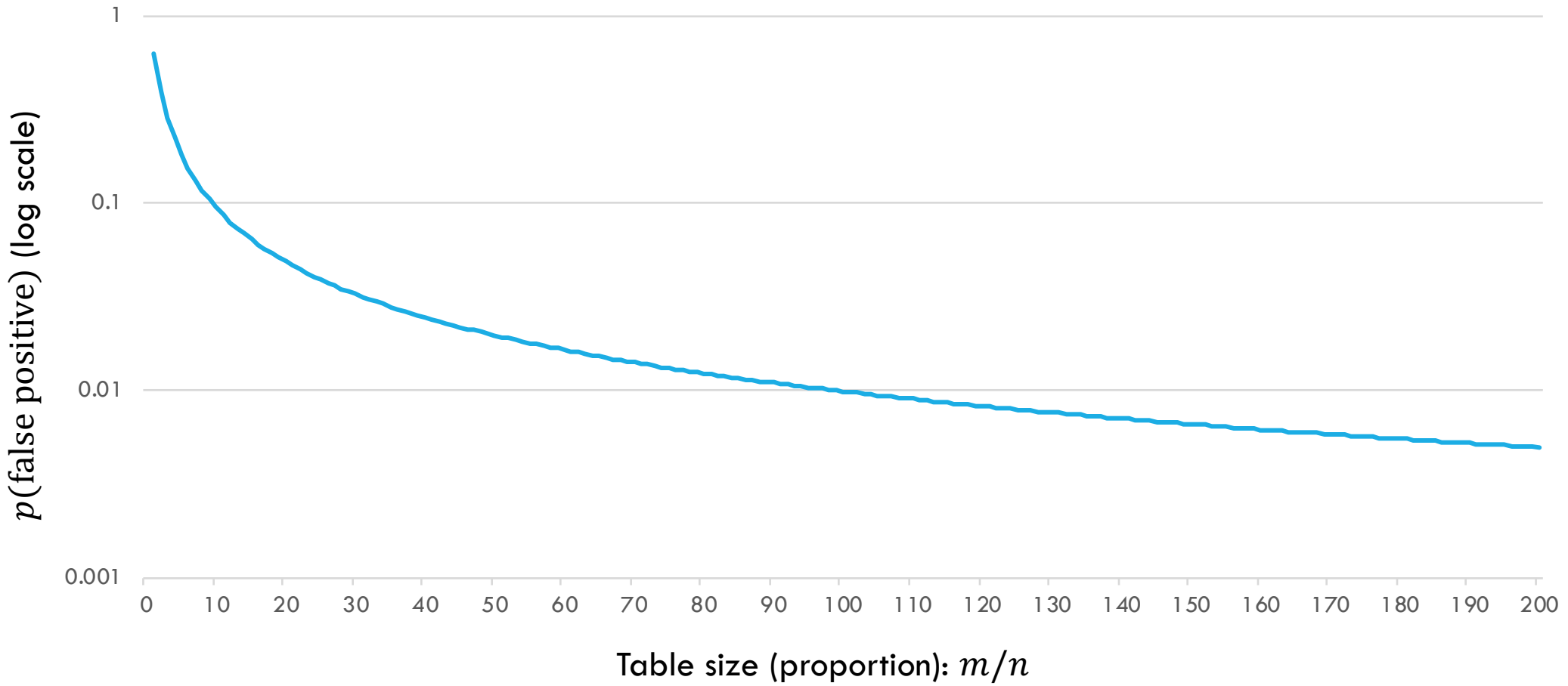
Then,

- $p(\text{false positive}) = 1 - e^{-\frac{n}{m}} \leq \hat{p}$
- $\frac{n}{m} \leq \ln\left(\frac{1}{1-\hat{p}}\right)$
- $m \geq \frac{n}{\ln\left(\frac{1}{1-\hat{p}}\right)}$
- So, $m \geq 99.5n$

If you wanted the derivation:

$$1 - e^{-\frac{n}{m}} \leq \hat{p}$$
$$e^{-\frac{n}{m}} \geq 1 - \hat{p}$$
$$-\frac{n}{m} \geq \log(1 - \hat{p})$$
$$\frac{n}{m} \leq \log\left(\frac{1}{1-\hat{p}}\right)$$
$$m \geq \frac{n}{\log\left(\frac{1}{1-\hat{p}}\right)}$$

PROBABILITY OF FALSE POSITIVE



SUMMARY SO FAR

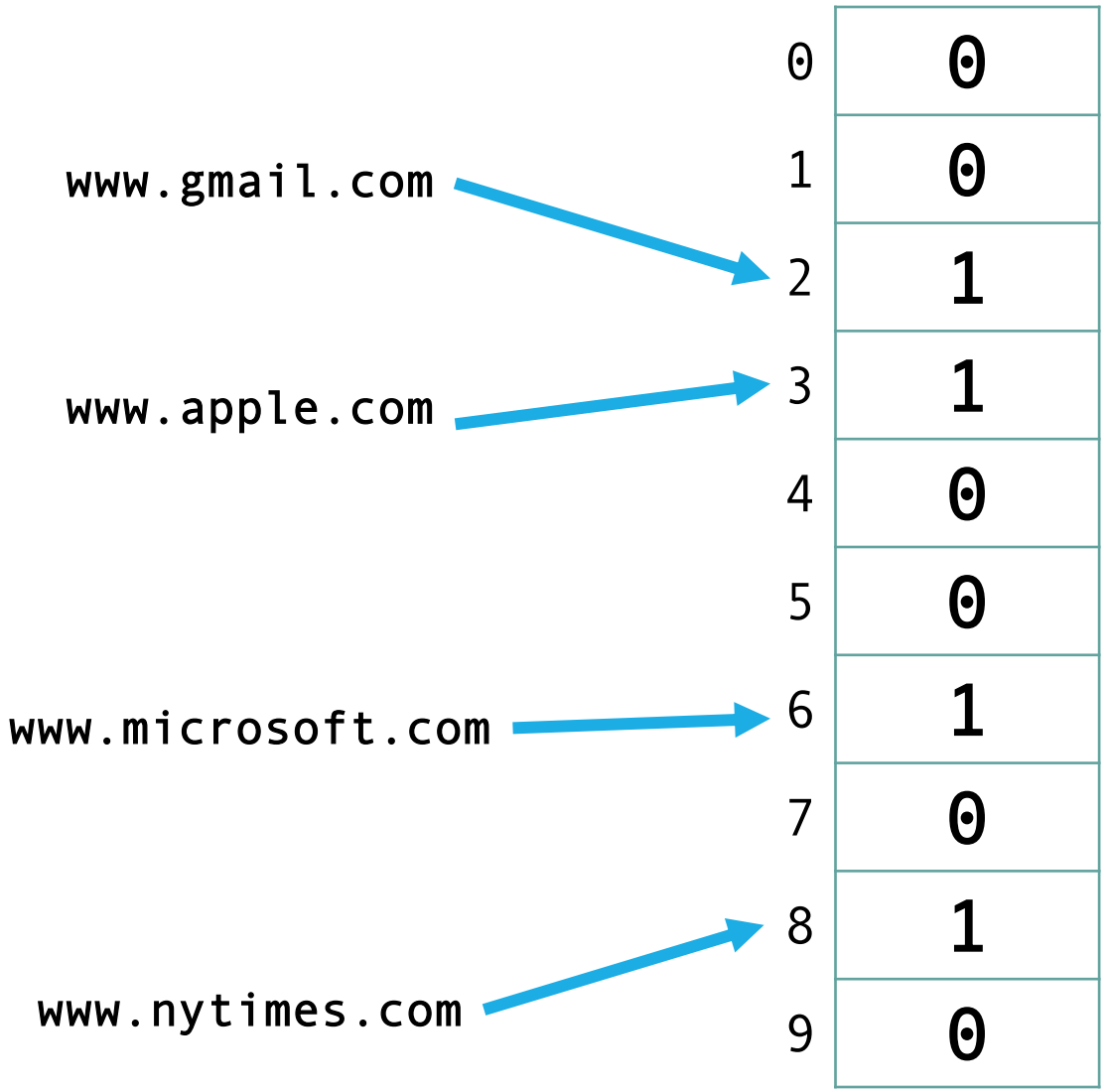
Only set k bits per item in a table of size m.

To start, let k = 1

- For each item, we only set 1 bit!

Reduced space but chance of false positives.

Can we do better?



BLOOM FILTER: SET K BITS PER ITEM

Use k hash functions

E.g., for $k = 2$, have h_1 and h_2

Insert x:

- Set the cells at <u>both</u> $h_1(x)$ and $h_2(x)$ to be 1

Lookup x:

- Return True only if cells at <u>both</u> $h_1(x)$ and $h_2(x)$ are 1

BLOOM FILTER: SET K BITS PER ITEM

Use k hash functions

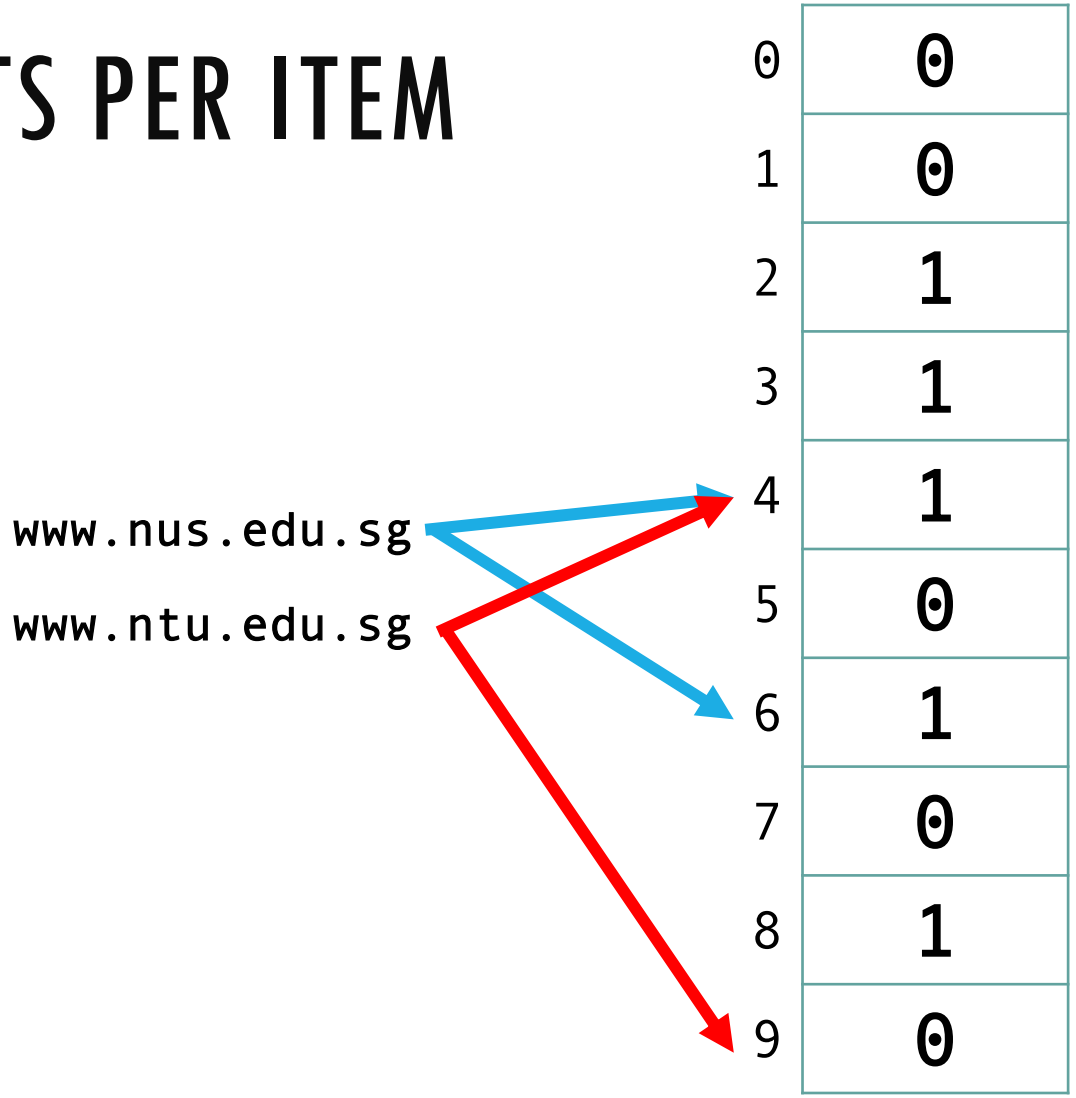
E.g., for $k = 2$, have h_1 and h_2

Insert x:

- Set the cells at <u>both</u> $h_1(x)$ and $h_2(x)$ to be 1

Lookup x:

- Return True only if cells at <u>both</u> $h_1(x)$ and $h_2(x)$ are 1

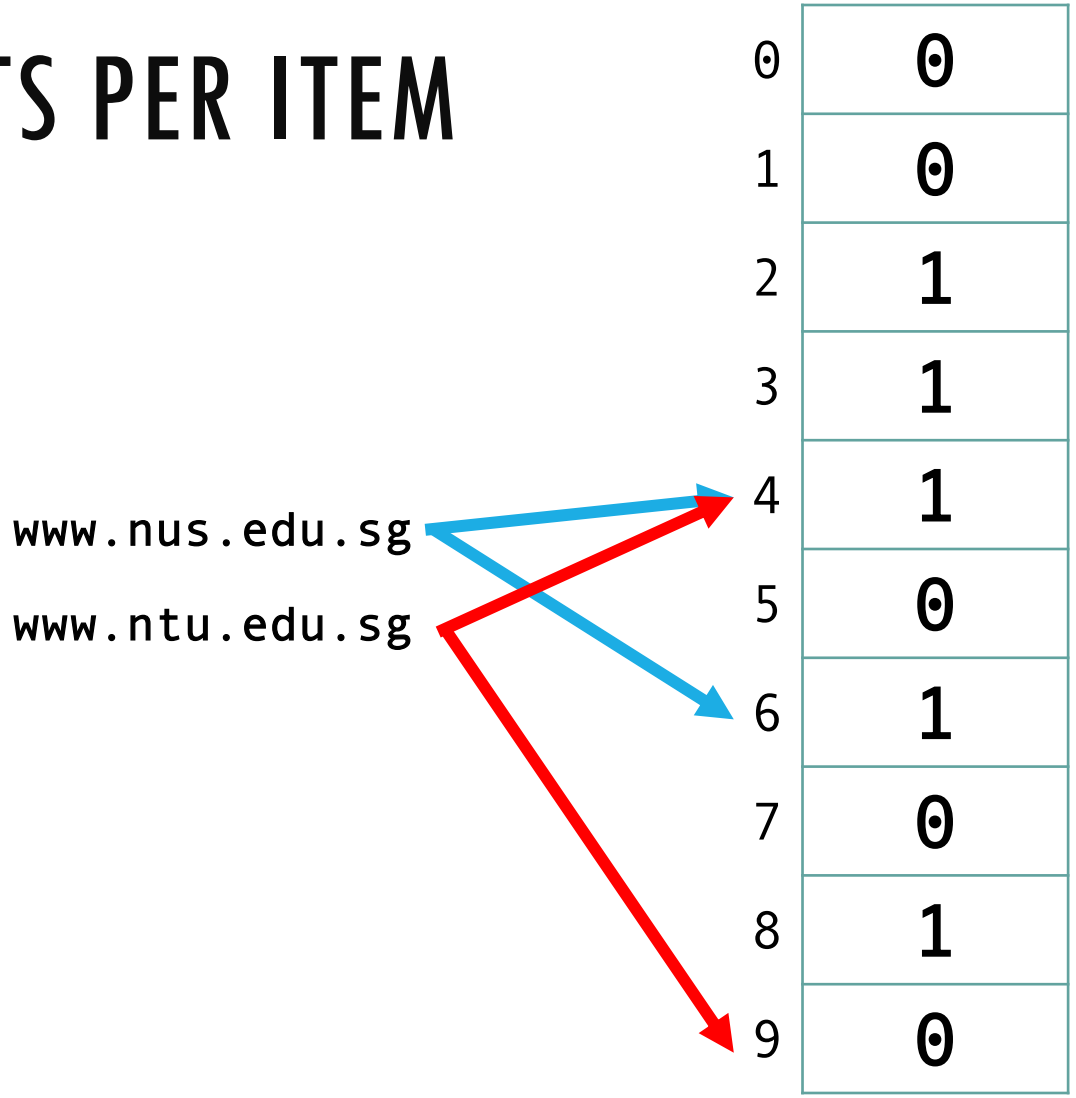


BLOOM FILTER: SET K BITS PER ITEM

Trade-offs:

- Require k collisions for a false positive
- Each item takes more "space"

What is the probability of a false positive?



RECALL: PROB. OF A FALSE POSITIVE

Table size is m, we store n elements

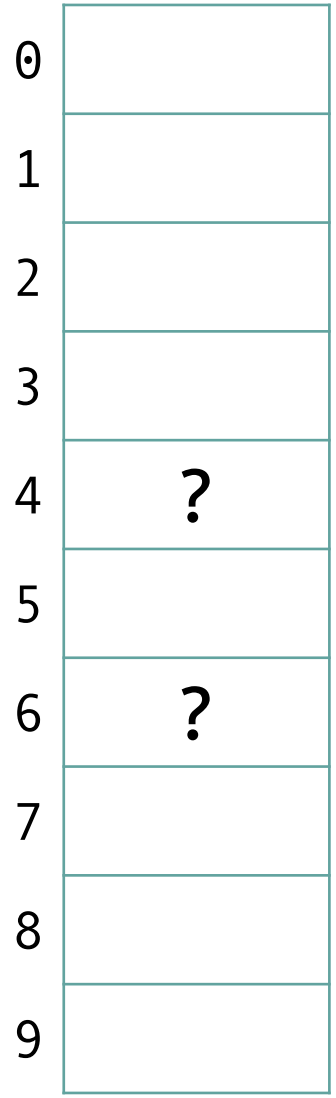
Set k bits per item

Test an item **not** in the set.

$$p(\text{false positive}) = p(\text{all cells are } 1)$$

$$p(\text{false positive}) = \underbrace{\big(1 - p(\text{cell is } 0)\big)^k}$$

Not strictly correct! Assumes independence for probabilities of each bit being set.



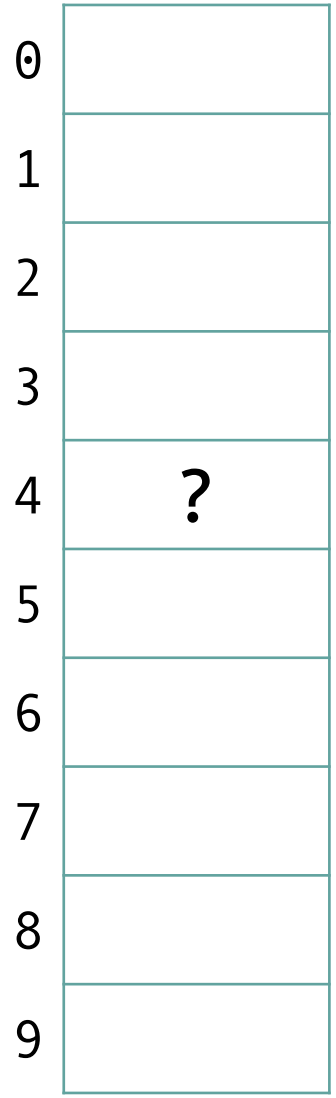
PROBABILITY OF A FALSE POSITIVE?

What is the probability that a cell is a 0?

- No previous insertion has set it to 1
- None of the previous k hash functions (per insert) set it to 1

Say $n = 1$ (only 1 item inserted in the table)

- Then $p(\text{cell is } 0) = \left(1 - \frac{1}{m}\right)^k$ (Simple uniform hashing assumption)



PROBABILITY OF A FALSE POSITIVE?

Say $n = 1$ (only 1 item inserted in the table)

- Then $p(\text{cell is } 0) = \left(1 - \frac{1}{m}\right)^{k}$ (Simple uniform hashing assumption)

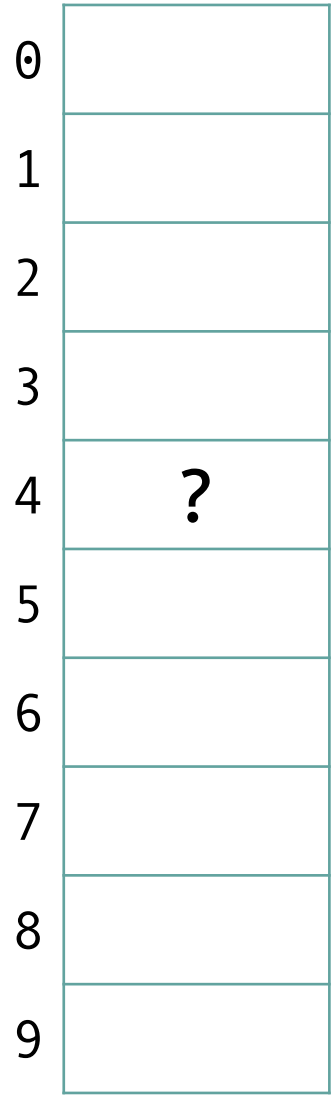
Say $n = 2$ (2 items inserted)

- Then $p(\text{cell is } 0) = \left(1 - \frac{1}{m}\right)^{2k}$ (Simple uniform hashing assumption)

…

For n items inserted

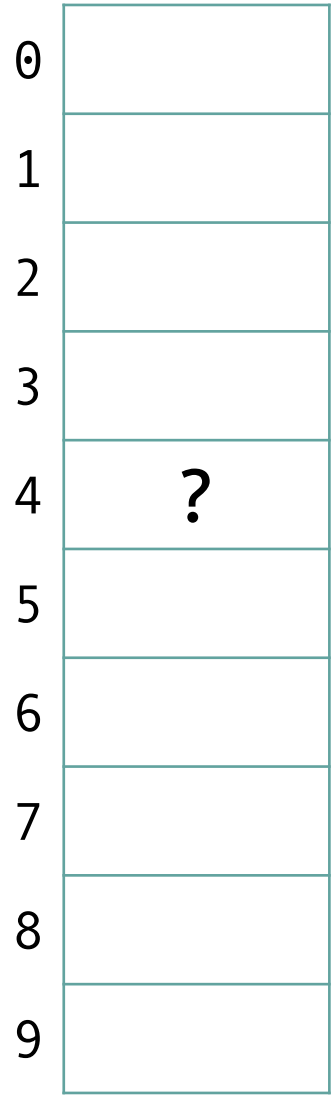
- Then $p(\text{cell is } 0) = \left(1 - \frac{1}{m}\right)^{nk}$ (Simple uniform hashing assumption)



PROBABILITY OF A FALSE POSITIVE?

For n items inserted

- Then $p(\text{cell is } 0) = \left(1 - \frac{1}{m}\right)^{kn} \approx e^{-\frac{kn}{m}}$



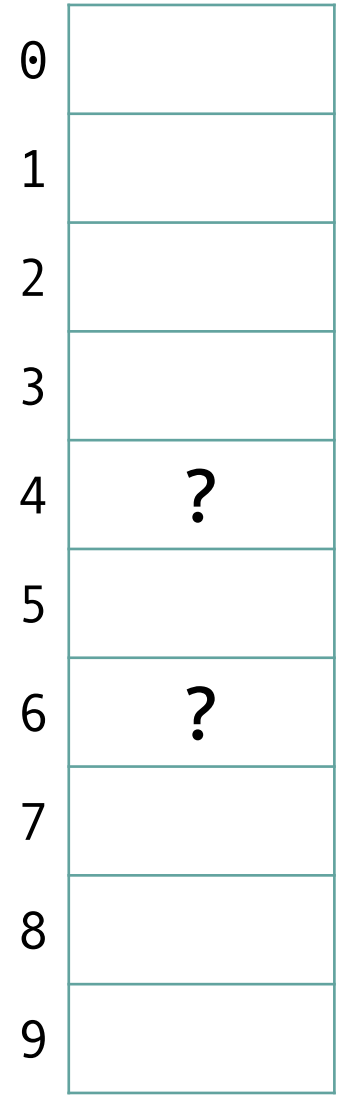
RECALL: PROB. OF A FALSE POSITIVE

Table size is m, we store n elements

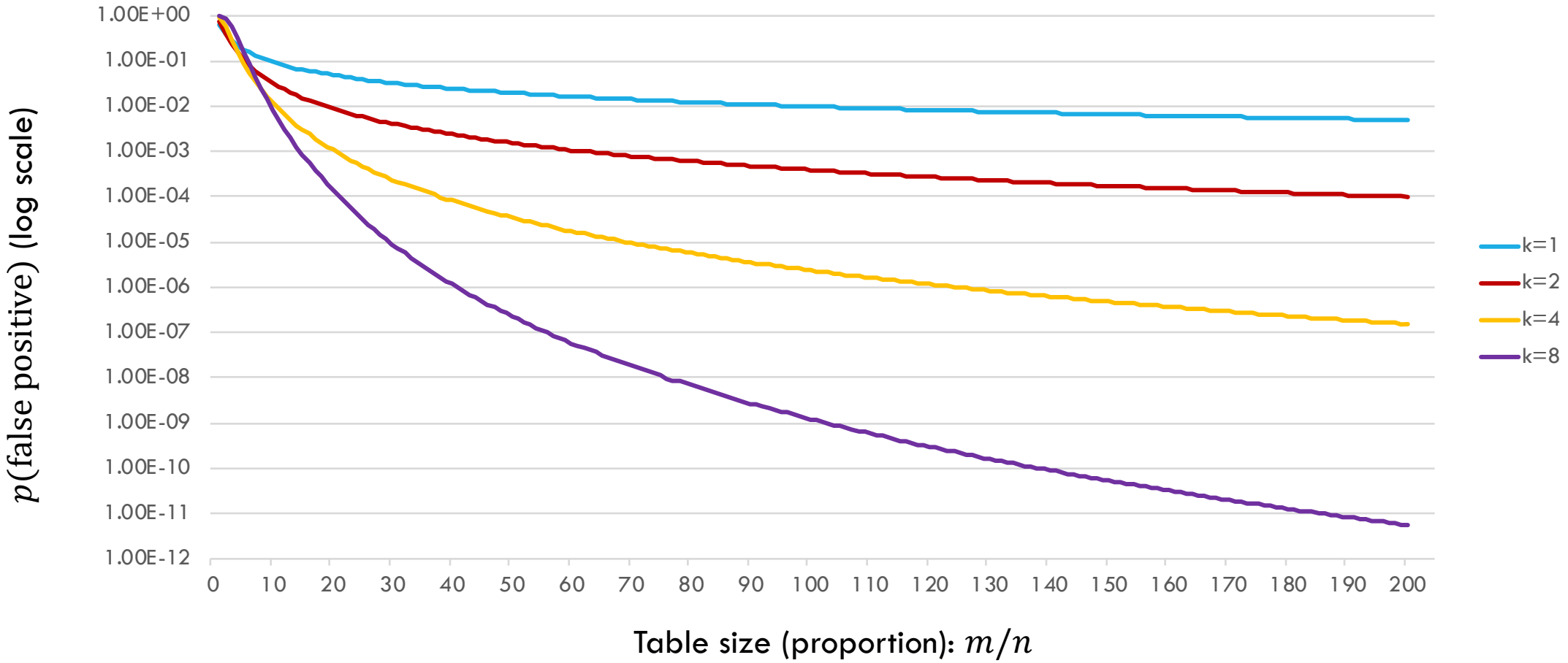
Set k bits per item

Test an item **not** in the set.

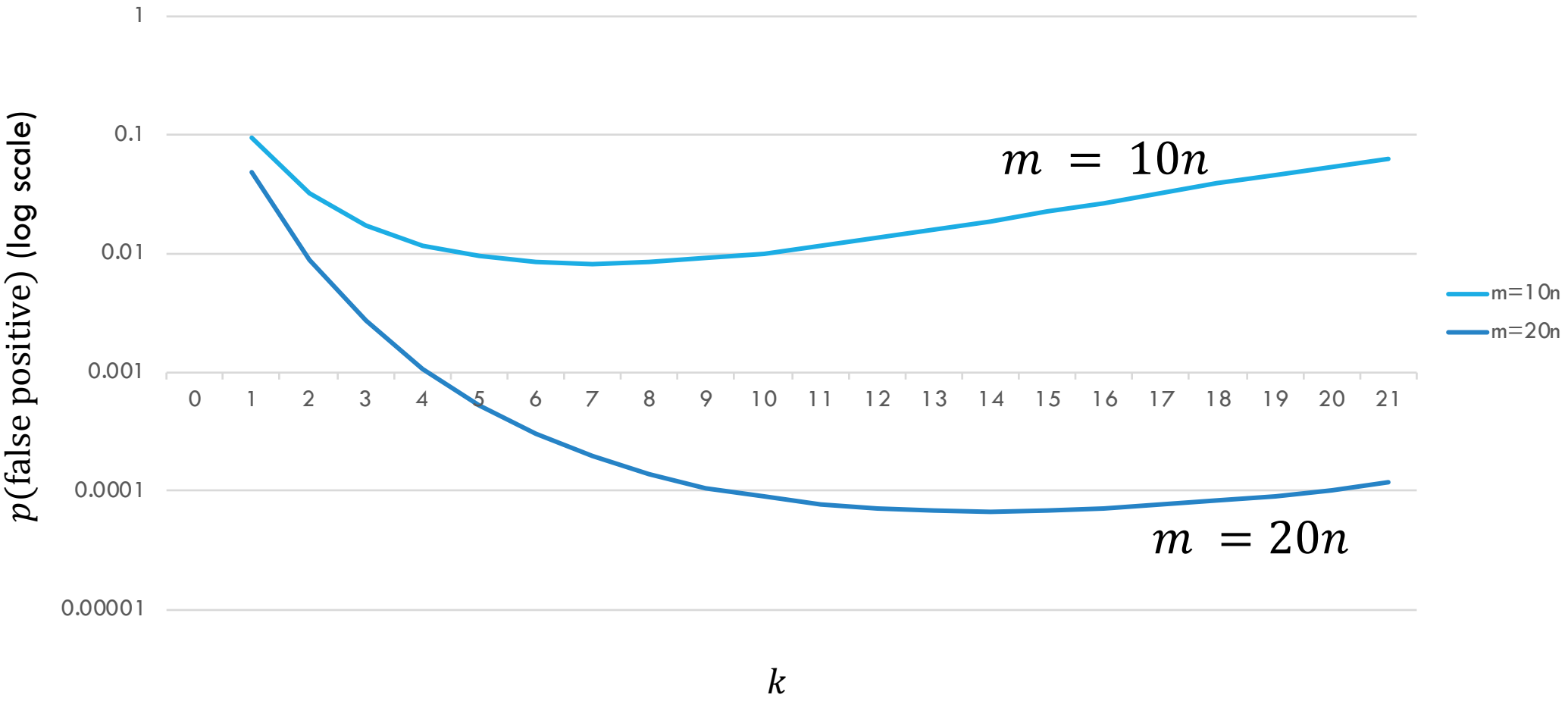
$$p(\text{false positive}) = p(\text{all cells are 1})$$
$$p(\text{false positive}) = \left(1 - p(\text{cell is 0})\right)^k$$
$$p(\text{false positive}) \approx \left(1 - e^{-\frac{kn}{m}}\right)^k$$



PROBABILITY OF FALSE POSITIVE



PROBABILITY OF FALSE POSITIVE VS K



OPTIMAL VALUE FOR K?

Probability of false positive:

- $p(\text{false positive}) \approx \left(1 - e^{-\frac{kn}{m}}\right)^k$

Choose k that minimizes $p(\text{false positive})$:

$$k = \frac{m}{n} \ln 2$$

SET DESIRED P(FALSE POSITIVE) AND N

Given some n items, and a desired false positive rate $\hat{p}$, can we find an optimal m?

Approximate solution:

- Use $k = \frac{m}{n} \ln 2$
- Substitute back into: $p = \left(1 - e^{-\left(\frac{m}{n} \ln 2 \frac{n}{m}\right)}\right)^{\frac{m}{n} \ln 2}$
- Simplify: $\ln p = -\frac{m}{n} (\ln 2)^2$
- So, $m = -\frac{n \log_2 p}{(\ln 2)^2}$

The size of Bloom filter is proportionate to the number of items n and the target false probability

SUMMARY: BLOOM FILTER

Use k hash functions

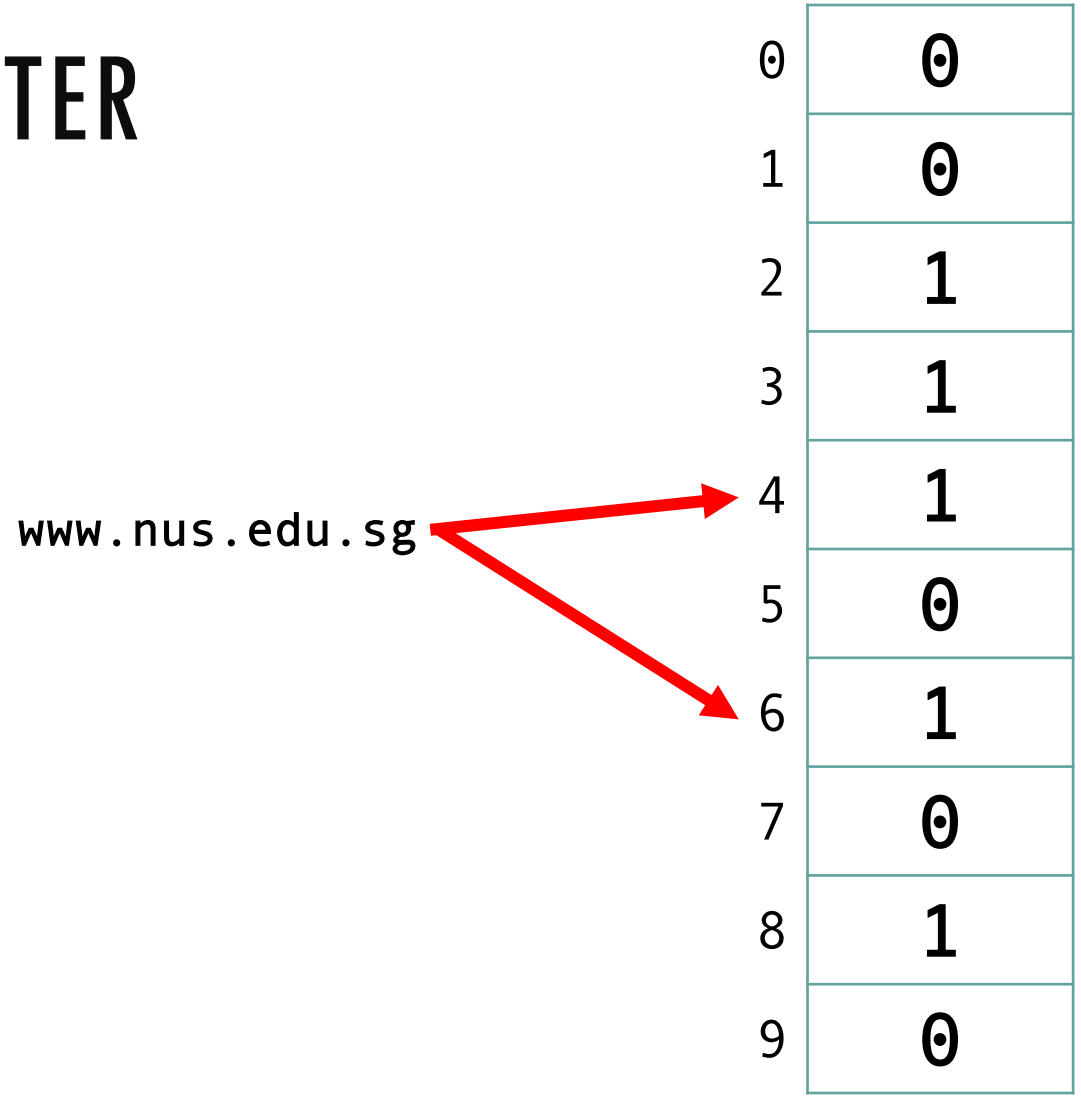
E.g., for $k = 2$, have h_1 and h_2

Insert x:

- Set the cells at <u>both</u> $h_1(x)$ and $h_2(x)$ to be 1

Lookup x:

- Return True only if cells at <u>both</u> $h_1(x)$ and $h_2(x)$ are 1



SUMMARY: BLOOM FILTER

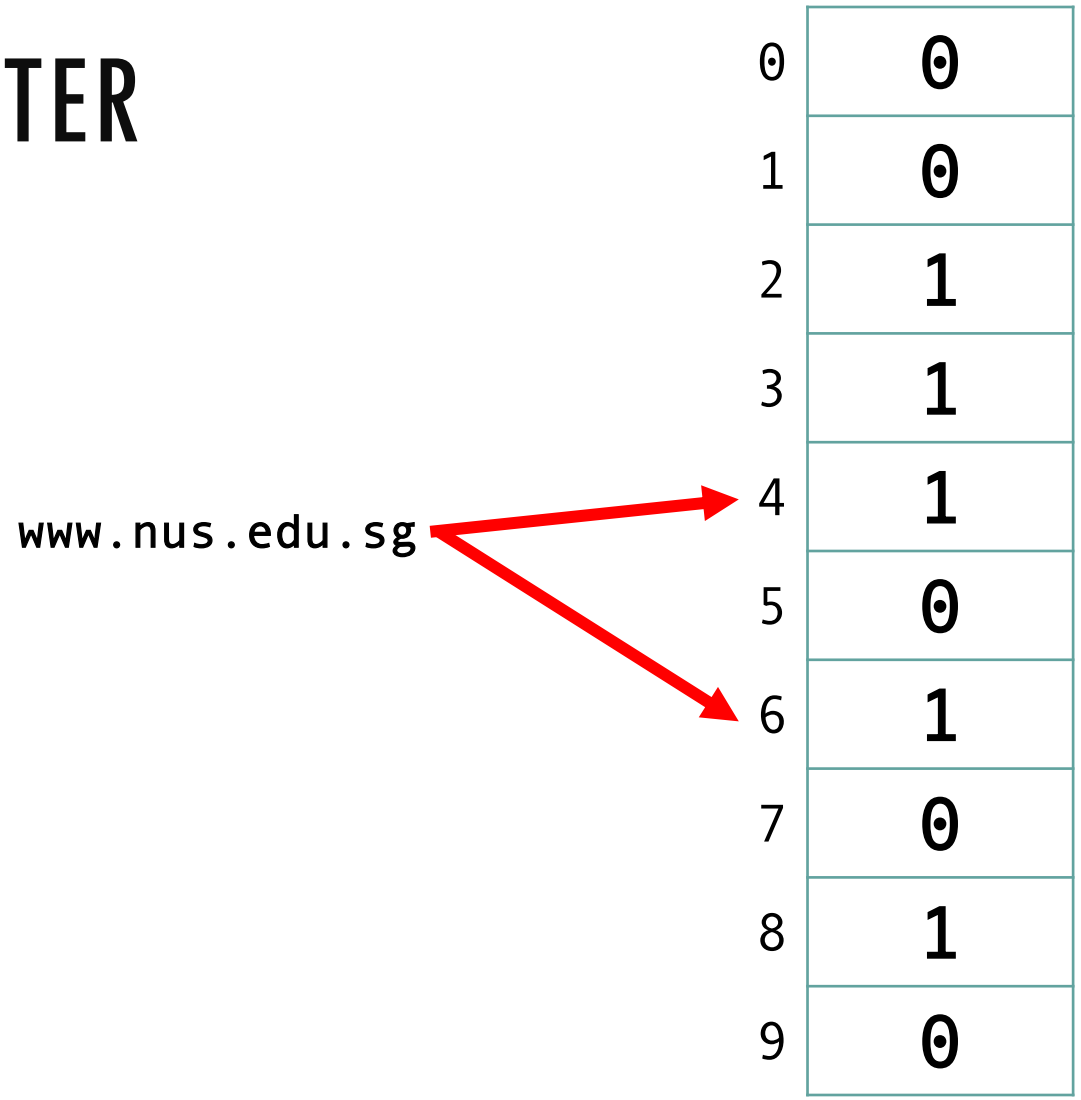
Use k hash functions

Saves space

But has probability of false positives.

Further reading: How about?

- **Deletions:** Can be handled using counters.
- **Intersections** and **Unions**.



QUESTIONS?

Poll Everywhere

https://bit.ly/2LvG9bq

CUCKOO HASHING

Did you know?

- Cuckoos lay eggs in other birds nests.
- When the cuckoo bird hatches, it pushes eggs/chicks out of the nest.

What a neat idea!

- Open addressing policy!
- Described by Rasmus Pagh and Flemming Friche Rodler in 2001.

CUCKOO HASHING: PERFORMANCE

Insertions seem to be quite complicated…

But: it takes expected $O(1)$ amortized time!

Analysis requires (a little) graph theory

To be continued in Week 12 … tomorrow!