Hashing and Sketching

Part Two

Outline for Today

- ***Recap from Last Time***
  - Where are we, again?
- ***Count Sketches***
  - A frequency estimator that shows off several key mathematical techniques.
- ***Cardinality Estimators***
  - How many different items have you seen?

Recap from Last Time

Distribution Property:
Each element should have an equal probability of being placed in each slot.

For any $x \in \mathscr{U}$ and random $h \in \mathcal{H}$, the value of $h(x)$ is uniform over $[m]$.

Independence Property:
Where one element is placed shouldn't impact where a second goes.

For any distinct $x, y \in \mathscr{U}$ and random $h \in \mathcal{H}$, $h(x)$ and $h(y)$ are independent random variables.

A family of hash functions $\mathscr{H}$ is called ***2-independent*** (or ***pairwise independent***) if it satisfies the distribution and independence properties.

Suppose there are two tunable values

$$\varepsilon \in (0, 1]$$
$$\delta \in (0, 1]$$

where ε represents ***accuracy*** and δ represents ***confidence***.

Goal: Make an estimator $\hat{A}$ for some quantity A where

With probability at least 1 – δ, } ***Probably***

$|\hat{A} - A| \leq \varepsilon \cdot size(input)$ } ***Approximately Correct***

for some measure of the size of the input.

What does it mean for an approximation to be "good"?

How to Build an Estimator

	Count-Min Sketch
Step One: Build a Simple Estimator	Hash items to counters; add +1 when item seen.
Step Two: Compute Expected Value of Estimator	Sum of indicators; 2-independent hashes have low collision rate.
Step Three: Apply Concentration Inequality	One-sided error; use expected value and Markov's inequality.
Step Four: Replicate to Boost Confidence	Take min; only fails if all estimates are bad.

New Stuff!

The Count Sketch

Frequency Estimation

- ***Recall:*** A frequency estimator is a data structure that supports
  - ***increment***(x), which increments the number of times that we've seen x, and
  - ***estimate***(x), which returns an estimate of how many times we've seen x.
- ***Notation:*** Assume that the elements we're processing are $x_1, \ldots, x_n$, and that the true frequency of element x_i is $\boldsymbol{a}_i$.
- Remember that the frequencies are not random variables – we're assuming that they're not under our control. Any randomness comes from hash functions.

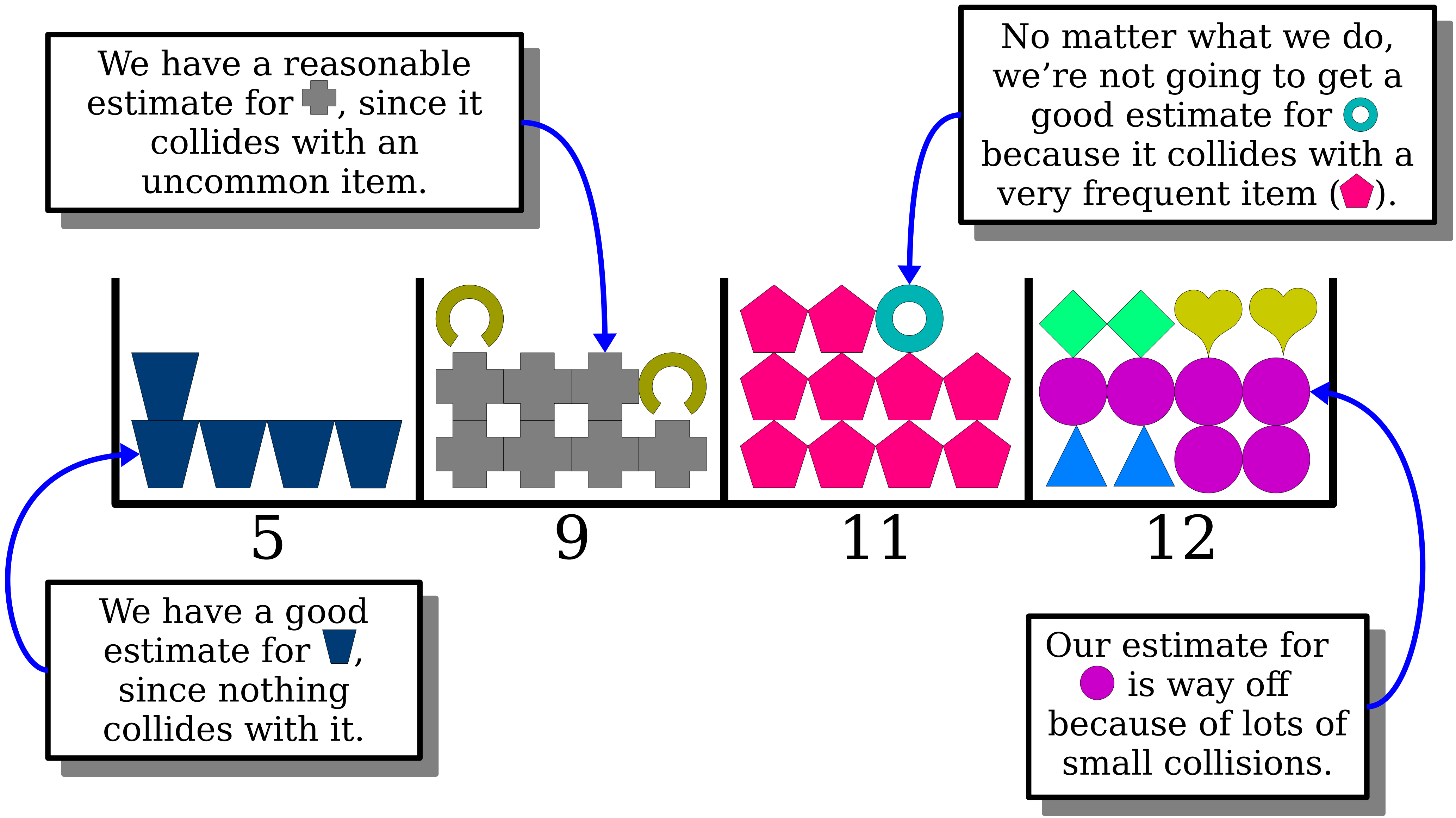
How to Build an Estimator

	Count-Min Sketch	***Count Sketch***
Step One: Build a Simple Estimator	Hash items to counters; add +1 when item seen.	
Step Two: Compute Expected Value of Estimator	Sum of indicators; 2-independent hashes have low collision rate.	
Step Three: Apply Concentration Inequality	One-sided error; use expected value and Markov's inequality.	
Step Four: Replicate to Boost Confidence	Take min; only fails if all estimates are bad.	

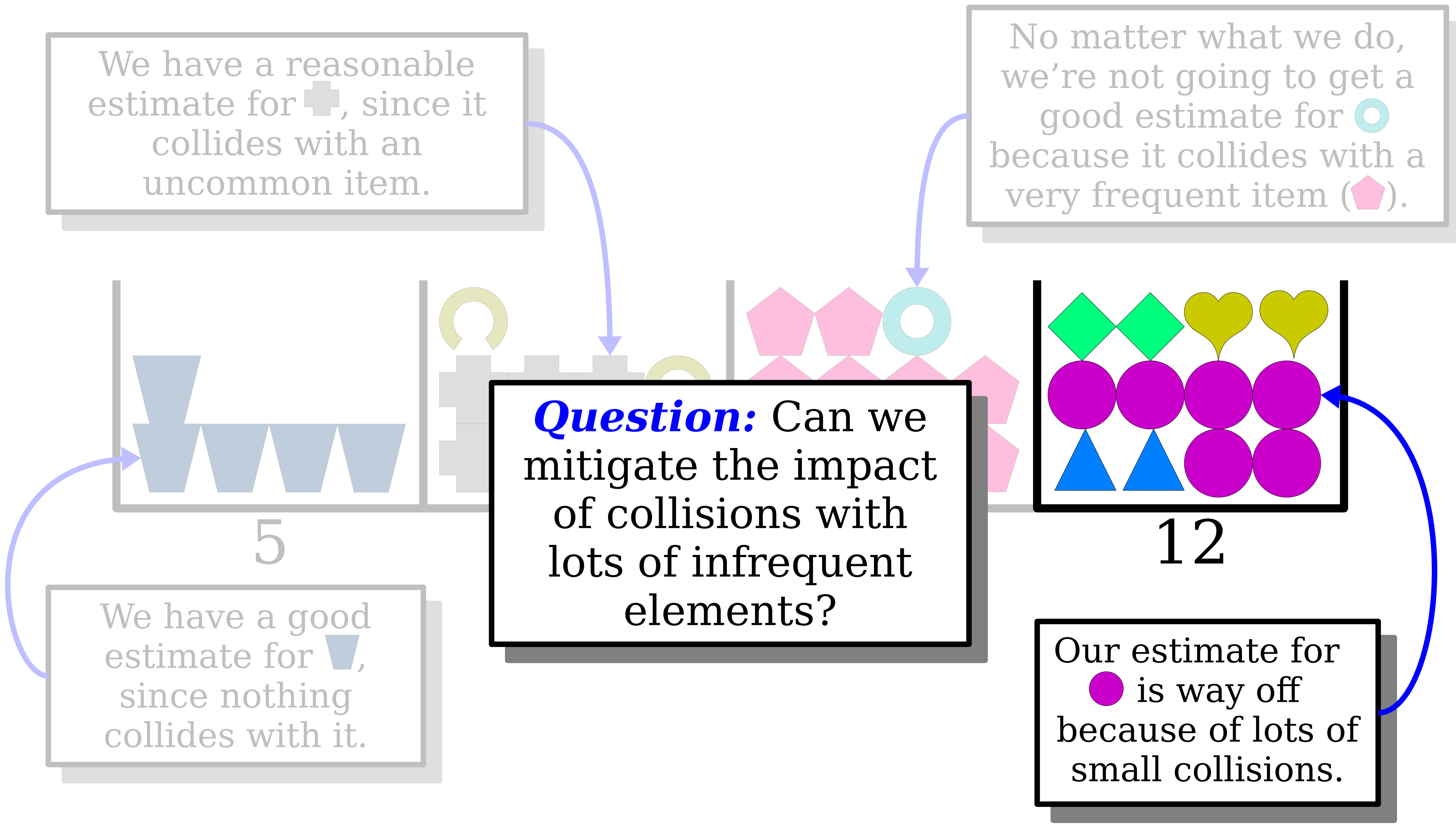
How to Build an Estimator

	Count-Min Sketch	**_Count Sketch_**
Step One: Build a Simple Estimator	Hash items to counters; add +1 when item seen.	
Step Two: Compute Expected Value of Estimator	Sum of indicators; 2-independent hashes have low collision rate.	
Step Three: Apply Concentration Inequality	One-sided error; use expected value and Markov's inequality.	
Step Four: Replicate to Boost Confidence	Take min; only fails if all estimates are bad.	

Revisiting Count-Min

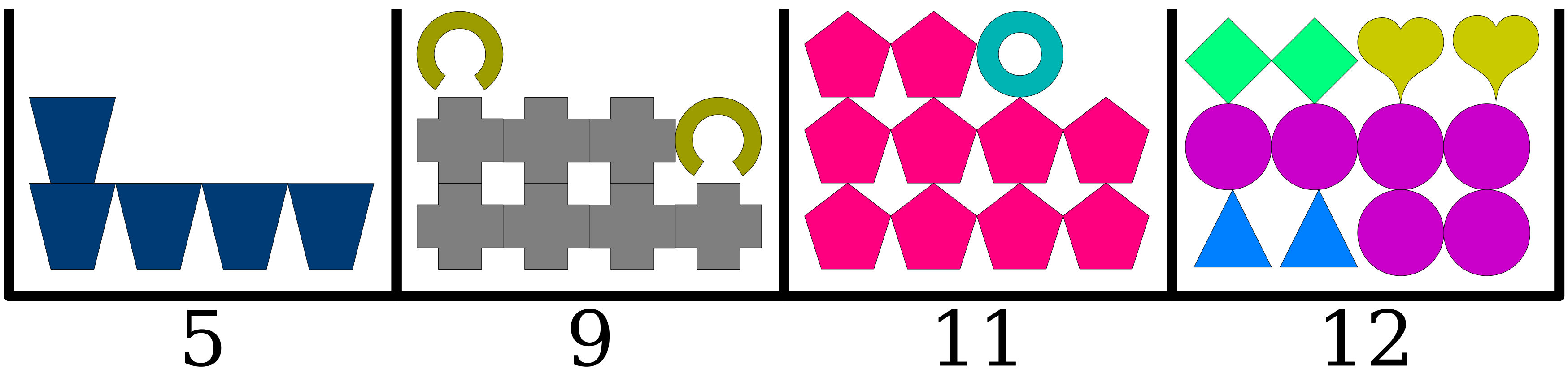


Revisiting Count-Min



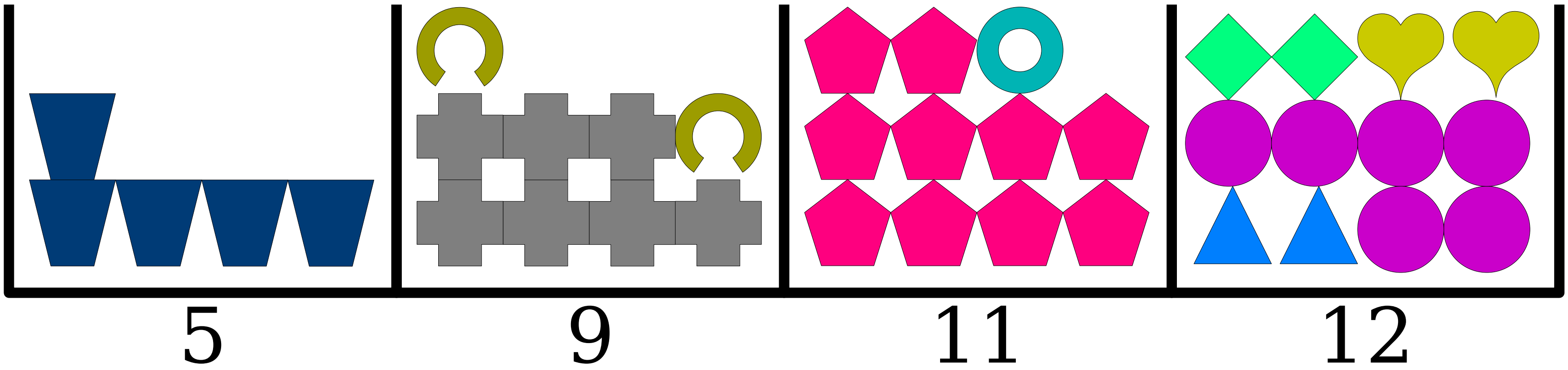
The Setup

- As before, create an array of counters and assign each item a counter.



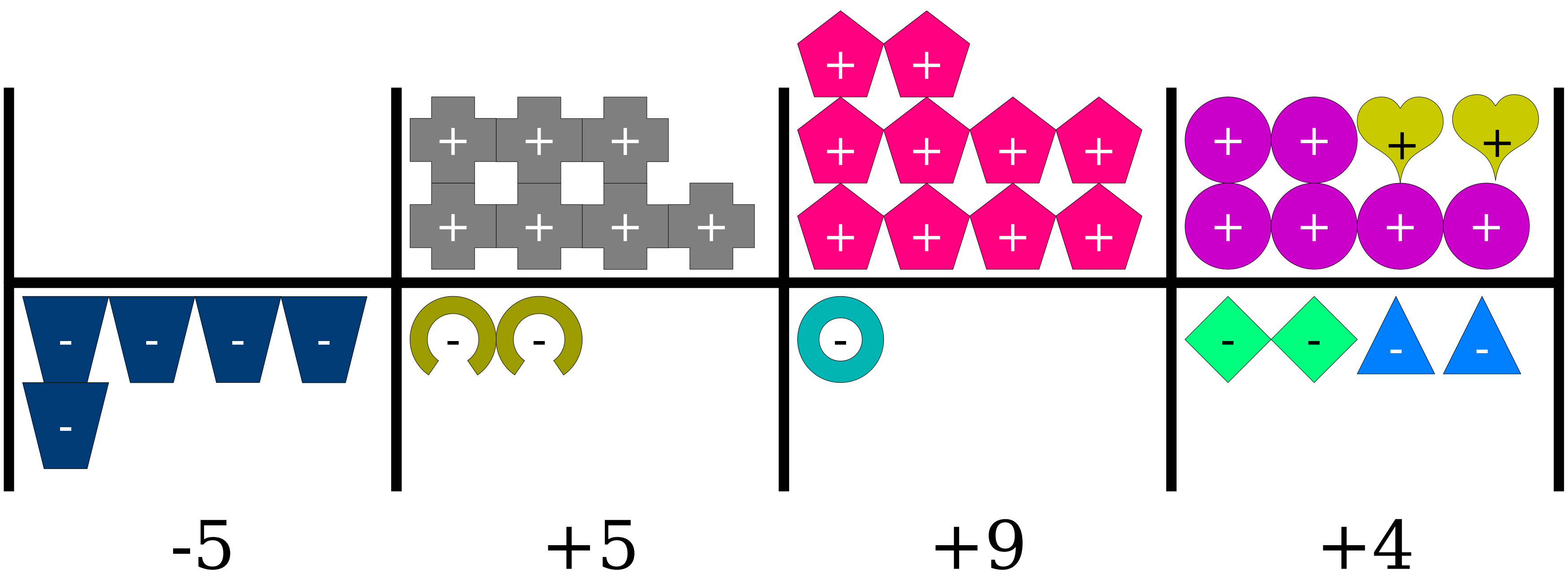
The Setup

- As before, create an array of counters and assign each item a counter.
- ***Key New Step:*** For each item x, assign x either +1 or -1.



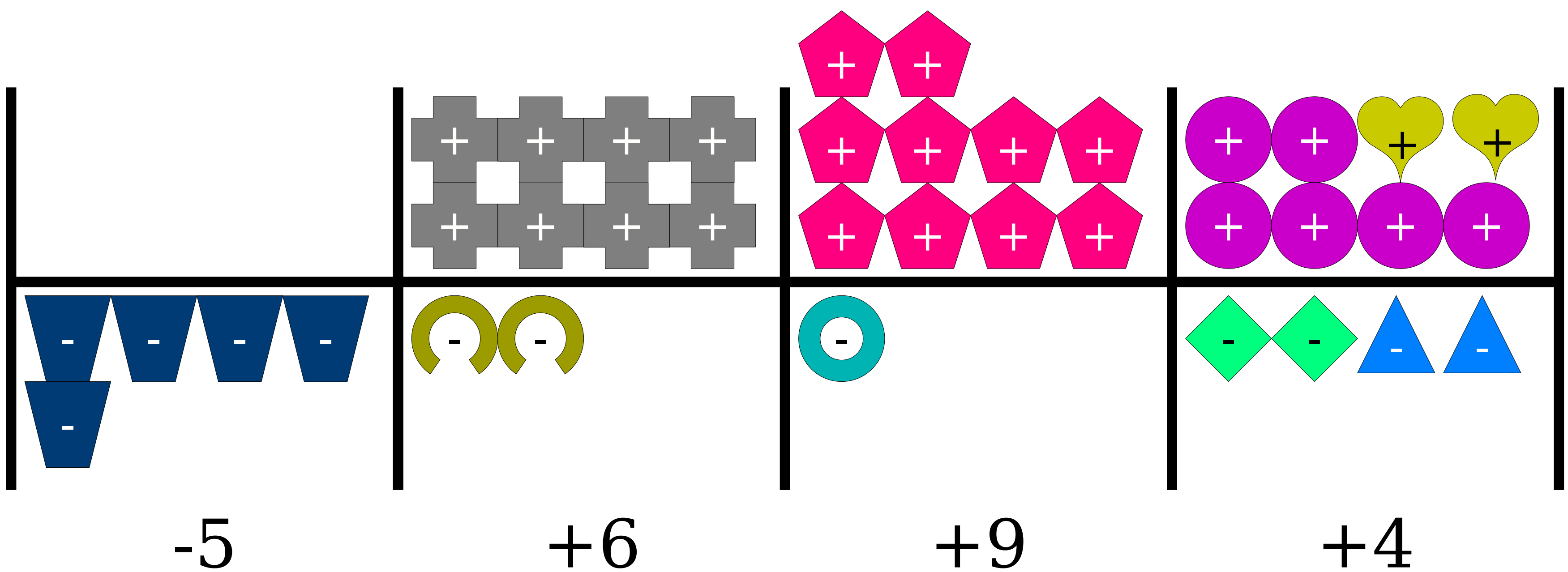
The Setup

- As before, create an array of counters and assign each item a counter.
- ***Key New Step:*** For each item x, assign x either +1 or -1.
  - To ***increment***(x), go to **`count`**[$h(x)$] and add ±1 as appropriate.
  - To ***estimate***(x), return **`count`**[$h(x)$], multiplied by ±1 as appropriate.



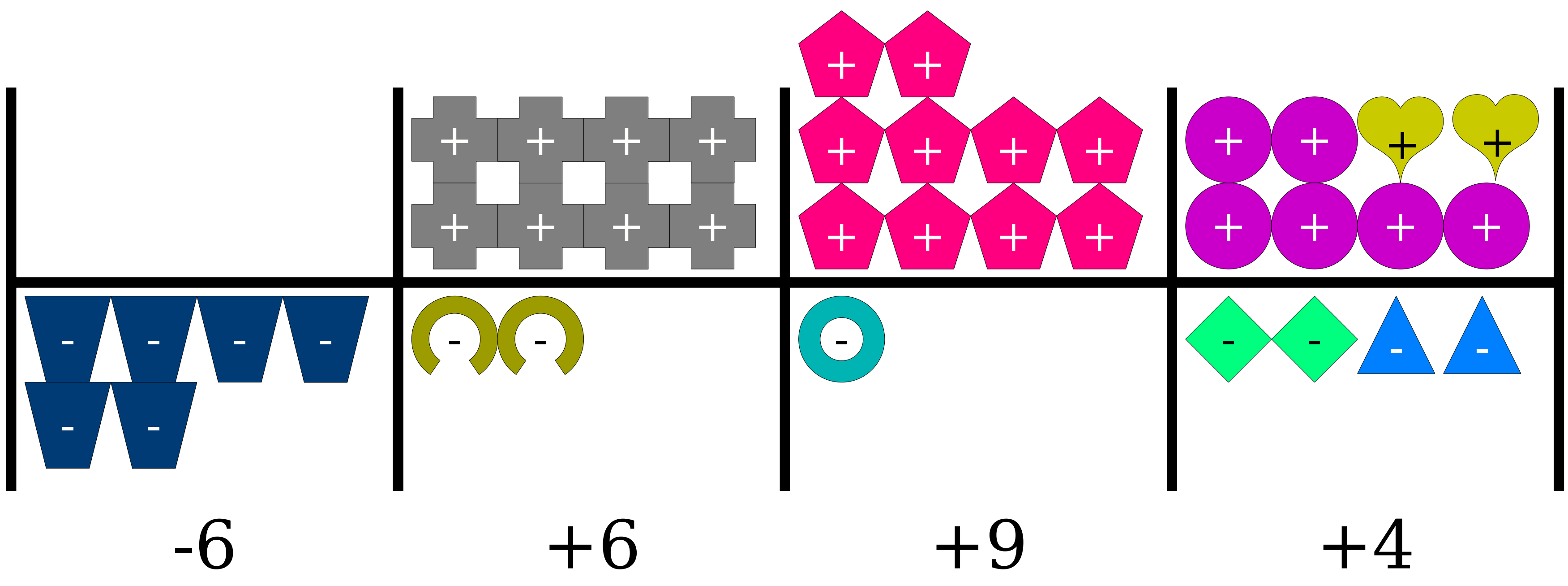
The Setup

- As before, create an array of counters and assign each item a counter.
- ***Key New Step:*** For each item x, assign x either +1 or -1.
  - To ***increment***(x), go to **`count`**[$h(x)$] and add ±1 as appropriate.
  - To ***estimate***(x), return **`count`**[$h(x)$], multiplied by ±1 as appropriate.



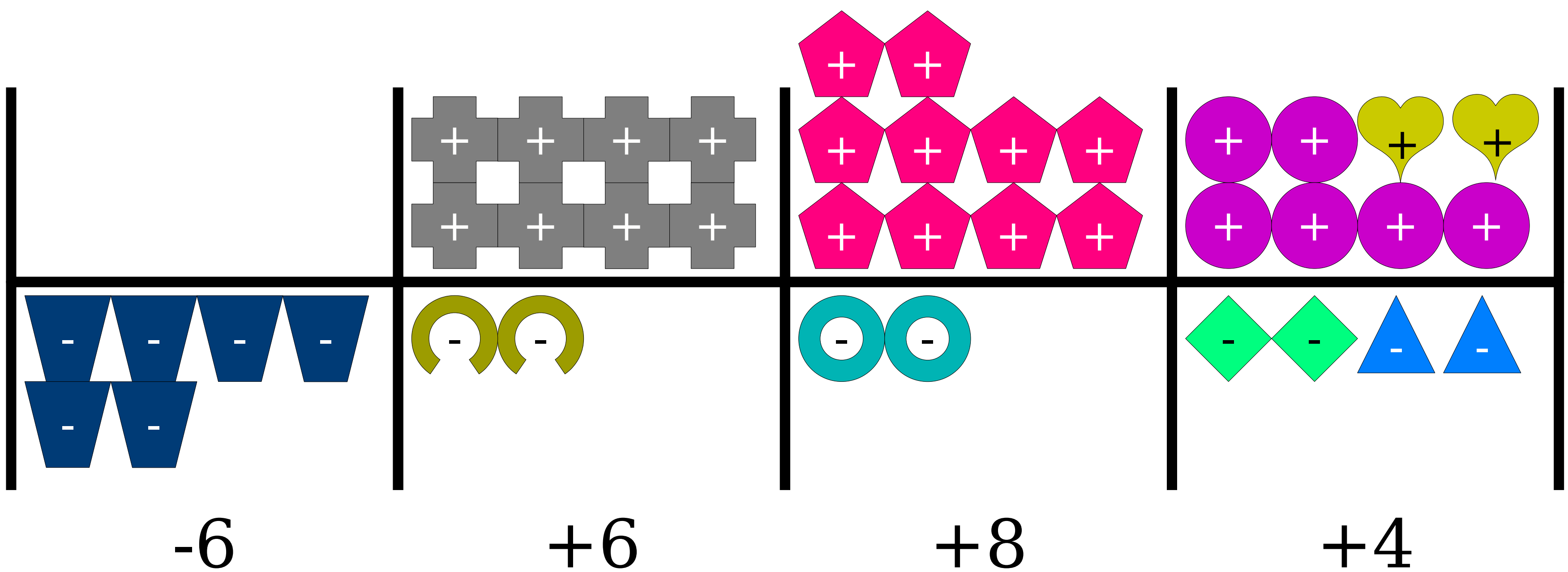
The Setup

- As before, create an array of counters and assign each item a counter.
- ***Key New Step:*** For each item x, assign x either +1 or -1.
  - To ***increment***(x), go to **`count`**[$h(x)$] and add ±1 as appropriate.
  - To ***estimate***(x), return **`count`**[$h(x)$], multiplied by ±1 as appropriate.

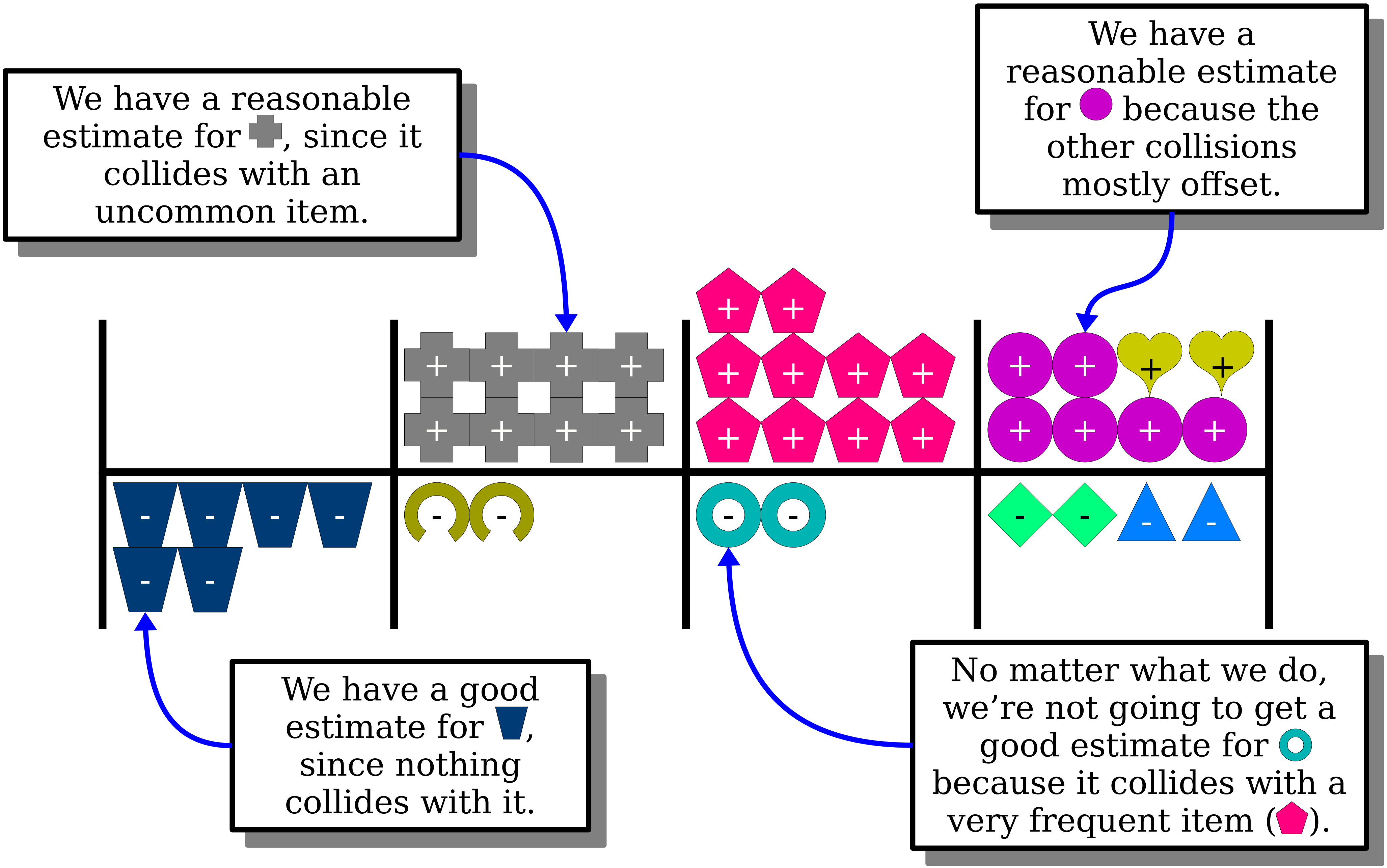


The Setup

- As before, create an array of counters and assign each item a counter.
- ***Key New Step:*** For each item x, assign x either +1 or -1.
  - To ***increment***(x), go to **count**[$h(x)$] and add ±1 as appropriate.
  - To ***estimate***(x), return **count**[$h(x)$], multiplied by ±1 as appropriate.

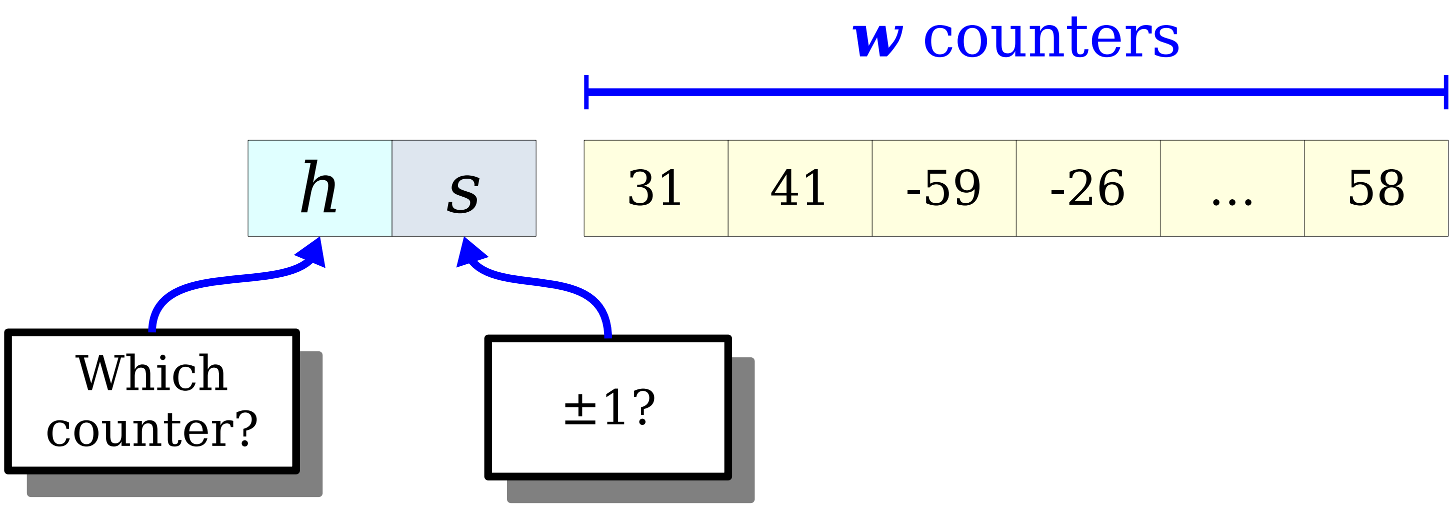


The Setup


We have a reasonable estimate for , since it collides with an uncommon item.
We have a reasonable estimate for because the other collisions mostly offset.
+
-
We have a good estimate for , since nothing collides with it.
No matter what we do, we're not going to get a good estimate for because it collides with a very frequent item ().

Formalizing This

- Maintain an array of counters of length w.
- Pick $h \in \mathscr{H}$ chosen uniformly at random from a 2-independent family of hash functions from $\mathscr{U}$ to w.
- Pick $s \in \mathscr{U}$ uniformly randomly and independently of h from a 2-independent family from $\mathscr{U}$ to {-1, +1}.
- ***increment***(x): **count**[$h(x)$] += $s(x)$.
- ***estimate***(x), return $s(x) \cdot$ **count**[$h(x)$].



How to Build an Estimator

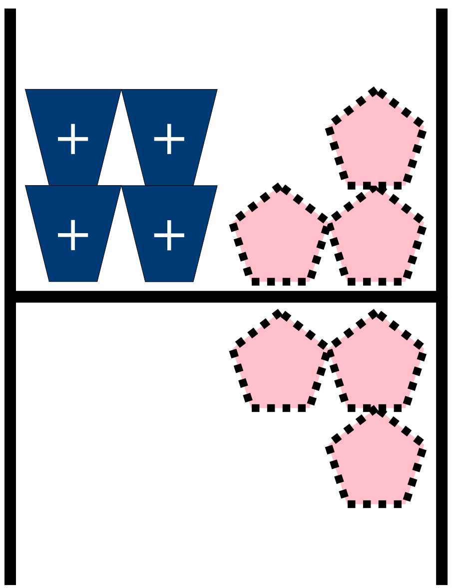
	Count-Min Sketch	***Count Sketch***
Step One: Build a Simple Estimator	Hash items to counters; add +1 when item seen.	Hash items to counters; add ±1 when item seen.
Step Two: Compute Expected Value of Estimator	Sum of indicators; 2-independent hashes have low collision rate.	
Step Three: Apply Concentration Inequality	One-sided error; use expected value and Markov's inequality.	
Step Four: Replicate to Boost Confidence	Take min; only fails if all estimates are bad.	

How to Build an Estimator

	Count-Min Sketch	***Count Sketch***
Step One: Build a Simple Estimator	Hash items to counters; add +1 when item seen.	Hash items to counters; add ±1 when item seen.
Step Two: Compute Expected Value of Estimator	Sum of indicators; 2-independent hashes have low collision rate.	
Step Three: Apply Concentration Inequality	One-sided error; use expected value and Markov's inequality.	
Step Four: Replicate to Boost Confidence	Take min; only fails if all estimates are bad.	

The Expectation, Intuitively

- Focus on any element x_i whose frequency we're estimating.
- Think about any element that collides with us.
- With 50% probability, it *increases* our estimate.
- With 50% probability, it *decreases* our estimate.
- ***Intuition:*** The expected value weights both options equally, so our estimator will be unbiased.



Formalizing the Intuition

- Define $\hat{\boldsymbol{a}}_i$ to be our estimate of $\boldsymbol{a}_i$.
- As before, $\hat{\boldsymbol{a}}_i$ will depend on how the other elements are distributed. Unlike before, it now also depends on signs given to the elements by s.
- Specifically, for each other x_j that collides with x_i, the estimate $\hat{\boldsymbol{a}}_i$ includes an error term of

$$s(x_i) \cdot s(x_j) \cdot \boldsymbol{a}_j$$

- Why?

Formulate a hypothesis, but ***don't post anything in chat just yet***.

Formalizing the Intuition

- Define $\hat{\boldsymbol{a}}_i$ to be our estimate of $\boldsymbol{a}_i$.
- As before, $\hat{\boldsymbol{a}}_i$ will depend on how the other elements are distributed. Unlike before, it now also depends on signs given to the elements by s.
- Specifically, for each other x_j that collides with x_i, the estimate $\hat{\boldsymbol{a}}_i$ includes an error term of

$$s(x_i) \cdot s(x_j) \cdot \boldsymbol{a}_j$$

- Why?

Now, ***private chat me your best guess***. Not sure? Just answer "??".

Formalizing the Intuition

- Define $\hat{\boldsymbol{a}}_i$ to be our estimate of $\boldsymbol{a}_i$.
- As before, $\hat{\boldsymbol{a}}_i$ will depend on how the other elements are distributed. Unlike before, it now also depends on signs given to the elements by s.
- Specifically, for each other x_j that collides with x_i, the estimate $\hat{\boldsymbol{a}}_i$ includes an error term of

$$s(x_i) \cdot s(x_j) \cdot \boldsymbol{a}_j$$

- Why?
  - The counter for x_i will have $s(x_j)\ \boldsymbol{a}_j$ added in.
  - We multiply the counter by $s(x_i)$ before returning it.

Formalizing the Intuition

- Define $\hat{\boldsymbol{a}}_i$ to be our estimate of $\boldsymbol{a}_i$.
- As before, $\hat{\boldsymbol{a}}_i$ will depend on how the other elements are distributed. Unlike before, it now also depends on signs given to the elements by s.
- Specifically, for each other x_j that collides with x_i, the estimate $\hat{\boldsymbol{a}}_i$ includes an error term of

$$s(x_i) \cdot s(x_j) \cdot \boldsymbol{a}_j$$

- Why?
  - If $s(x_i)$ and $s(x_j)$ point in the same direction, the terms add to the total.
  - If $s(x_i)$ and $s(x_j)$ point in different directions, the terms subtract from the total.

Formalizing the Intuition

- In our quest to learn more about $\hat{\boldsymbol{a}}_i$, let's have X_j be a random variable indicating whether x_i and x_j collided with one another:

$$X_j = \begin{cases} 1 & \text{if } h(x_i) = h(x_j) \\ 0 & \text{if } h(x_i) \neq h(x_j) \end{cases}$$

Formalizing the Intuition

- In our quest to learn more about $\hat{\boldsymbol{a}}_i$, let's have X_j be a random variable indicating whether x_i and x_j collided with one another:

$$X_j = \begin{cases} 1 & \text{if } h(x_i) = h(x_j) \\ 0 & \text{if } h(x_i) \neq h(x_j) \end{cases}$$

- We can then express $\hat{\boldsymbol{a}}_i$ in terms of the signed contributions from the items x_i collides with:

$$\hat{\boldsymbol{a}}_i = \sum_j \boldsymbol{a}_j s(x_i) s(x_j) X_j$$

Formalizing the Intuition

- In our quest to learn more about $\hat{\boldsymbol{a}}_i$, let's have X_j be a random variable indicating whether x_i and x_j collided with one another:

$$X_j = \begin{cases} 1 & \text{if } h(x_i) = h(x_j) \\ 0 & \text{if } h(x_i) \neq h(x_j) \end{cases}$$

- We can then express $\hat{\boldsymbol{a}}_i$ in terms of the signed contributions from the items x_i collides with:

$$\hat{\boldsymbol{a}}_i = \sum_j \boldsymbol{a}_j s(x_i) s(x_j) X_j$$

This is how much the collision impacts our estimate.

Formalizing the Intuition

- In our quest to learn more about $\hat{\boldsymbol{a}}_i$, let's have X_j be a random variable indicating whether x_i and x_j collided with one another:

$$X_j = \begin{cases} 1 & \text{if } h(x_i) = h(x_j) \\ 0 & \text{if } h(x_i) \neq h(x_j) \end{cases}$$

- We can then express $\hat{\boldsymbol{a}}_i$ in terms of the signed contributions from the items x_i collides with:

$$\hat{\boldsymbol{a}}_i = \sum_j \boldsymbol{a}_j s(x_i) s(x_j) X_j$$

This is how much the collision impacts our estimate.

We only care about items we collided with.

Formalizing the Intuition

- In our quest to learn more about $\hat{\boldsymbol{a}}_i$, let's have X_j be a random variable indicating whether x_i and x_j collided with one another:

$$X_j = \begin{cases} 1 & \text{if } h(x_i) = h(x_j) \\ 0 & \text{if } h(x_i) \neq h(x_j) \end{cases}$$

- We can then express $\hat{\boldsymbol{a}}_i$ in terms of the signed contributions from the items x_i collides with:

$$\hat{\boldsymbol{a}}_i = \sum_j \boldsymbol{a}_j s(x_i) s(x_j) X_j = \boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j$$

This is how much the collision impacts our estimate.

We only care about items we collided with.

$$\mathrm{E}[\hat{\boldsymbol{a}}_i] = \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j]$$

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j]
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}\Big[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j\Big] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}\Big[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j\Big]
\end{aligned}
$$

Hey, it’s linearity of expectation!

$$\begin{aligned} \mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\ &= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\ &= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \end{aligned}$$

$$
\begin{aligned}
E[\hat{\boldsymbol{a}}_i] &= E[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= E[\boldsymbol{a}_i] + E[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} E[\boldsymbol{a}_j s(x_i) s(x_j) X_j]
\end{aligned}
$$

Remember that $\boldsymbol{a}_i$ and the like aren't random variables.

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j]
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j]
\end{aligned}
$$

$$
\boxed{X_j = \begin{cases} 1 & \text{if } h(x_i) = h(x_j) \\ 0 & \text{if } h(x_i) \neq h(x_j) \end{cases}}
$$

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j]
\end{aligned}
$$

We chose the hash functions h and s independently of one another.

$$
X_j = \begin{cases} 1 & \text{if } h(x_i) = h(x_j) \\ 0 & \text{if } h(x_i) \neq h(x_j) \end{cases}
$$

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i) s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j]
\end{aligned}
$$

We chose the hash functions h and s independently of one another.

$$
X_j = \begin{cases} 1 & \text{if } h(x_i) = h(x_j) \\ 0 & \text{if } h(x_i) \neq h(x_j) \end{cases}
$$

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i) s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j]
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i) s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i)] \mathrm{E}[s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j]
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i) s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i)] \mathrm{E}[s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j]
\end{aligned}
$$

Since s is drawn from a 2-independent family of hash functions, we know $s(x_i)$ and $s(x_j)$ are independent random variables.

$$
\begin{aligned}
E[\hat{\boldsymbol{a}}_i] &= E[\boldsymbol{a}_i + \sum_{j\neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= E[\boldsymbol{a}_i] + E[\sum_{j\neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j\neq i} E[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j\neq i} E[s(x_i) s(x_j)] E[\boldsymbol{a}_j X_j] \\
&= \boldsymbol{a}_i + \sum_{j\neq i} E[s(x_i)] E[s(x_j)] E[\boldsymbol{a}_j X_j]
\end{aligned}
$$

$E[s(x_i)] =$

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i) s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i)] \mathrm{E}[s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j]
\end{aligned}
$$

$\mathrm{E}[s(x_i)] =$

s is drawn from a 2-independent family of hash functions.

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i) s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i)] \mathrm{E}[s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j]
\end{aligned}
$$

$\mathrm{E}[s(x_i)] =$

s is drawn from a 2-independent family of hash functions.

$s(x_i)$ is uniform over {-1, +1}

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i) s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i)] \mathrm{E}[s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j]
\end{aligned}
$$

$\mathrm{E}[s(x_i)] =$

s is drawn from a 2-independent family of hash functions.

$s(x_i)$ is uniform over {-1, +1}

$\Pr[s(x_i) = -1] = ½$ $\Pr[s(x_i) = +1] = ½$

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i) s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i)] \mathrm{E}[s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j]
\end{aligned}
$$

$\mathrm{E}[s(x_i)] = ½ \cdot (-1) + ½ \cdot (+1)$

s is drawn from a 2-independent family of hash functions.

$s(x_i)$ is uniform over $\{-1, +1\}$

$\Pr[s(x_i) = -1] = ½$ $\Pr[s(x_i) = +1] = ½$

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i) s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i)] \mathrm{E}[s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j]
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}[s(x_i)] &= \tfrac{1}{2} \cdot (-1) + \tfrac{1}{2} \cdot (+1) \\
&= 0
\end{aligned}
$$

s is drawn from a 2-independent family of hash functions.

$s(x_i)$ is uniform over $\{-1, +1\}$

$\Pr[s(x_i) = -1] = \tfrac{1}{2}$ $\Pr[s(x_i) = +1] = \tfrac{1}{2}$

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i) s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i)] \mathrm{E}[s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} 0
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}[s(x_i)] &= \tfrac{1}{2} \cdot (-1) + \tfrac{1}{2} \cdot (+1) \\
&= 0
\end{aligned}
$$

s is drawn from a 2-independent family of hash functions.

$s(x_i)$ is uniform over $\{-1, +1\}$

$\Pr[s(x_i) = -1] = \tfrac{1}{2}$ $\quad$ $\Pr[s(x_i) = +1] = \tfrac{1}{2}$

$$
\begin{aligned}
\mathrm{E}[\hat{\boldsymbol{a}}_i] &= \mathrm{E}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{E}[\boldsymbol{a}_i] + \mathrm{E}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i) s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} \mathrm{E}[s(x_i)] \mathrm{E}[s(x_j)] \mathrm{E}[\boldsymbol{a}_j X_j] \\
&= \boldsymbol{a}_i + \sum_{j \neq i} 0 \\
&= \boldsymbol{a}_i
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}[s(x_i)] &= ½ \cdot (-1) + ½ \cdot (+1) \\
&= 0
\end{aligned}
$$

s is drawn from a 2-independent family of hash functions.

$s(x_i)$ is uniform over $\{-1, +1\}$

$\Pr[s(x_i) = -1] = ½$ $\Pr[s(x_i) = +1] = ½$

How to Build an Estimator

	Count-Min Sketch	***Count Sketch***
Step One: Build a Simple Estimator	Hash items to counters; add +1 when item seen.	Hash items to counters; add ±1 when item seen.
Step Two: Compute Expected Value of Estimator	Sum of indicators; 2-independent hashes have low collision rate.	2-independence breaks up products; ±1 variables have zero expected value.
Step Three: Apply Concentration Inequality	One-sided error; use expected value and Markov's inequality.	
Step Four: Replicate to Boost Confidence	Take min; only fails if all estimates are bad.	

How to Build an Estimator

	Count-Min Sketch	***Count Sketch***
Step One: Build a Simple Estimator	Hash items to counters; add +1 when item seen.	Hash items to counters; add ±1 when item seen.
Step Two: Compute Expected Value of Estimator	Sum of indicators; 2-independent hashes have low collision rate.	2-independence breaks up products; ±1 variables have zero expected value.
Step Three: Apply Concentration Inequality	One-sided error; use expected value and Markov's inequality.	
Step Four: Replicate to Boost Confidence	Take min; only fails if all estimates are bad.	

A Hitch

- In the count-min sketch, we used Markov's inequality to bound the probability that we get a bad estimate.
- This worked because we had a ***one-sided error***: the distance $\hat{a}_i - a_i$ from the true answer was nonnegative.
- With the count sketch, we have a ***two-sided error***: $\hat{a}_i - a_i$ can be negative in the count sketch because collisions can *decrease* the estimate $\hat{a}_i$ below the true value a_i.
- We'll need to use a different technique to bound the error.

Chebyshev to the Rescue

- ***Chebyshev's inequality*** states that for any random variable X with finite variance, given any $c > 0$, we have

$$\Pr\left[\ |X - \mathrm{E}[X]| \geq c\ \right] \leq \frac{\mathrm{Var}[X]}{c^2}.$$

- If we can get the variance of $\hat{\boldsymbol{a}}_i$, we can bound the probability that we get a bad estimate with our data structure.

$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] \quad = \quad \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j]$$

$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] = \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j]$$

$$\boxed{\mathrm{Var}[a + X] = \mathrm{Var}[X]}$$

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j]
\end{aligned}
$$

$$\boxed{\mathrm{Var}[a + X] = \mathrm{Var}[X]}$$

$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] = \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j]$$

$$= \boxed{\mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j]}$$

In general, Var is *not* a linear operator.

However, if the terms in the sum are ***pairwise uncorrelated***, then Var is linear.

Lemma: The terms in this sum are uncorrelated. *(Prove this!)*

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j]
\end{aligned}
$$

In general, Var is *not* a linear operator.

However, if the terms in the sum are ***pairwise uncorrelated***, then Var is linear.

Lemma: The terms in this sum are uncorrelated. *(Prove this!)*

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j]
\end{aligned}
$$

$$\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j]
\end{aligned}$$

The “Sum-o’-Var” Samovar!

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j]
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j]
\end{aligned}
$$

$$
\boxed{
\begin{aligned}
\mathrm{Var}[Z] &= \mathrm{E}[Z^2] - \mathrm{E}[Z]^2 \\
&\leq \mathrm{E}[Z^2]
\end{aligned}
}
$$

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&\leq \sum_{j \neq i} \mathrm{E}[(\boldsymbol{a}_j s(x_i) s(x_j) X_j)^2]
\end{aligned}
$$

$$
\boxed{\begin{aligned} \mathrm{Var}[Z] &= \mathrm{E}[Z^2] - \mathrm{E}[Z]^2 \\ &\leq \mathrm{E}[Z^2] \end{aligned}}
$$

$$\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&\leq \sum_{j \neq i} \mathrm{E}[(\boldsymbol{a}_j s(x_i) s(x_j) X_j)^2] \\
&= \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j^2 s(x_i)^2 s(x_j)^2 X_j^2]
\end{aligned}$$

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}\Big[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j\Big] \\
&= \mathrm{Var}\Big[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j\Big] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&\leq \sum_{j \neq i} \mathrm{E}[(\boldsymbol{a}_j s(x_i) s(x_j) X_j)^2] \\
&= \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j^2 s(x_i)^2 s(x_j)^2 X_j^2]
\end{aligned}
$$

$s(x) = \pm 1$,

so

$s(x)^2 = 1$

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&\leq \sum_{j \neq i} \mathrm{E}[(\boldsymbol{a}_j s(x_i) s(x_j) X_j)^2] \\
&= \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j^2 s(x_i)^2 s(x_j)^2 X_j^2] \\
&= \sum_{j \neq i} \boldsymbol{a}_j^2 \mathrm{E}[X_j^2]
\end{aligned}
$$

$s(x) = \pm 1$,

so

$s(x)^2 = 1$

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&\leq \sum_{j \neq i} \mathrm{E}[(\boldsymbol{a}_j s(x_i) s(x_j) X_j)^2] \\
&= \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j^2 s(x_i)^2 s(x_j)^2 X_j^2] \\
&= \sum_{j \neq i} \boldsymbol{a}_j^2 \mathrm{E}[X_j^2]
\end{aligned}
$$

$$
\boxed{X_j = \begin{cases} 1 & \text{if } h(x_i) = h(x_j) \\ 0 & \text{if } h(x_i) \neq h(x_j) \end{cases}}
$$

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&\leq \sum_{j \neq i} \mathrm{E}[(\boldsymbol{a}_j s(x_i) s(x_j) X_j)^2] \\
&= \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j^2 s(x_i)^2 s(x_j)^2 X_j^2] \\
&= \sum_{j \neq i} \boldsymbol{a}_j^2 \mathrm{E}[X_j^2]
\end{aligned}
$$

$$
\boxed{X_j^{\mathbf{2}} = \begin{cases} 1^{\mathbf{2}} & \text{if } h(x_i) = h(x_j) \\ 0^{\mathbf{2}} & \text{if } h(x_i) \neq h(x_j) \end{cases}}
$$

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&\leq \sum_{j \neq i} \mathrm{E}[(\boldsymbol{a}_j s(x_i) s(x_j) X_j)^2] \\
&= \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j^2 s(x_i)^2 s(x_j)^2 X_j^2] \\
&= \sum_{j \neq i} \boldsymbol{a}_j^2 \mathrm{E}[X_j^2]
\end{aligned}
$$

$$
\boxed{X_j^{\mathbf{2}} = \begin{cases} 1 & \text{if } h(x_i) = h(x_j) \\ 0 & \text{if } h(x_i) \neq h(x_j) \end{cases}}
$$

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&\leq \sum_{j \neq i} \mathrm{E}[(\boldsymbol{a}_j s(x_i) s(x_j) X_j)^2] \\
&= \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j^2 s(x_i)^2 s(x_j)^2 X_j^2] \\
&= \sum_{j \neq i} \boldsymbol{a}_j^2 \mathrm{E}[X_j^2]
\end{aligned}
$$

Useful Fact: If X is an indicator, then $X^2 = X$.

$$
X_j^{2} = \begin{cases} 1 & \text{if } h(x_i) = h(x_j) \\ 0 & \text{if } h(x_i) \neq h(x_j) \end{cases}
$$

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j] \\
&\leq \sum_{j \neq i} \mathrm{E}[(\boldsymbol{a}_j s(x_i) s(x_j) \boldsymbol{X}_j)^2] \\
&= \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j^2 s(x_i)^2 s(x_j)^2 \boldsymbol{X}_j^2] \\
&= \sum_{j \neq i} \boldsymbol{a}_j^2 \mathrm{E}[X_j^2] \\
&= \sum_{j \neq i} \boldsymbol{a}_j^2 \mathrm{E}[X_j]
\end{aligned}
$$

Useful Fact: If X is an indicator, then $X^2 = X$.

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&\leq \sum_{j \neq i} \mathrm{E}[(\boldsymbol{a}_j s(x_i) s(x_j) X_j)^2] \\
&= \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j^2 s(x_i)^2 s(x_j)^2 X_j^2] \\
&= \sum_{j \neq i} \boldsymbol{a}_j^2 \mathrm{E}[X_j^2] \\
&= \sum_{j \neq i} \boldsymbol{a}_j^2 \mathrm{E}[X_j]
\end{aligned}
$$

$$
\boxed{X_j = \begin{cases} 1 & \text{if } h(x_i) = h(x_j) \\ 0 & \text{if } h(x_i) \neq h(x_j) \end{cases}}
$$

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&\leq \sum_{j \neq i} \mathrm{E}[(\boldsymbol{a}_j s(x_i) s(x_j) X_j)^2] \\
&= \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j^2 s(x_i)^2 s(x_j)^2 X_j^2] \\
&= \sum_{j \neq i} \boldsymbol{a}_j^2 \mathrm{E}[X_j^2] \\
&= \sum_{j \neq i} \boldsymbol{a}_j^2 \mathrm{E}[X_j] \\
&= \frac{1}{w} \sum_{j \neq i} \boldsymbol{a}_j^2
\end{aligned}
$$

$$
\boxed{X_j = \begin{cases} 1 & \text{if } h(x_i) = h(x_j) \\ 0 & \text{if } h(x_i) \neq h(x_j) \end{cases}}
$$

$$
\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{a}}_i] &= \mathrm{Var}[\boldsymbol{a}_i + \sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \mathrm{Var}[\sum_{j \neq i} \boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&= \sum_{j \neq i} \mathrm{Var}[\boldsymbol{a}_j s(x_i) s(x_j) X_j] \\
&\leq \sum_{j \neq i} \mathrm{E}[(\boldsymbol{a}_j s(x_i) s(x_j) X_j)^2] \\
&= \sum_{j \neq i} \mathrm{E}[\boldsymbol{a}_j^2 s(x_i)^2 s(x_j)^2 X_j^2] \\
&= \sum_{j \neq i} \boldsymbol{a}_j^2 \mathrm{E}[X_j^2] \\
&= \sum_{j \neq i} \boldsymbol{a}_j^2 \mathrm{E}[X_j] \\
&= \frac{1}{w} \sum_{j \neq i} \boldsymbol{a}_j^2
\end{aligned}
$$

I know this might look really dense, but many of these substeps end up being really useful techniques. These ideas generalize, I promise.

Think of [$\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$, …] as a vector.

What does the following quantity represent?

$$\sum_j \boldsymbol{a}_j^2$$

$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] \;\leq\; \frac{1}{w}\sum_{j\neq i} \boldsymbol{a}_j^2$$

Think of [$\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$, …] as a vector.

What does the following quantity represent?

$$\sum_j \boldsymbol{a}_j^2$$

This is the square of the magnitude of the vector!

$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] \leq \frac{1}{w} \sum_{j \neq i} \boldsymbol{a}_j^2$$

Think of [$\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$, …] as a vector.

What does the following quantity represent?

$$\sum_j \boldsymbol{a}_j^2$$

This is the square of the magnitude of the vector!

The magnitude of a vector is called its ***L_2 norm*** and is denoted $\|\boldsymbol{a}\|_2$.

$$\|\boldsymbol{a}\|_2 = \sqrt{\sum_j \boldsymbol{a}_j^2}$$

$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] \leq \frac{1}{w} \sum_{j \neq i} \boldsymbol{a}_j^2$$

Think of [$\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$, …] as a vector.

What does the following quantity represent?

$$\sum_j \boldsymbol{a}_j^2$$

This is the square of the magnitude of the vector!

The magnitude of a vector is called its ***L_2* norm** and is denoted $\|\boldsymbol{a}\|_2$.

$$\|\boldsymbol{a}\|_2 = \sqrt{\sum_j \boldsymbol{a}_j^2}$$

Therefore, our above sum is $\|\boldsymbol{a}\|_2^2$.

$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] \leq \frac{1}{w} \sum_{j \neq i} \boldsymbol{a}_j^2$$

Think of [$\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$, …] as a vector.

What does the following quantity represent?

$$\sum_j \boldsymbol{a}_j^2$$

This is the square of the magnitude of the vector!

The magnitude of a vector is called its ***L_2* norm** and is denoted $\|\boldsymbol{a}\|_2$.

$$\|\boldsymbol{a}\|_2 = \sqrt{\sum_j \boldsymbol{a}_j^2}$$

Therefore, our above sum is $\|\boldsymbol{a}\|_2^2$.

$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] \leq \frac{1}{w}\sum_{j \neq i} \boldsymbol{a}_j^2 \leq \frac{\|\boldsymbol{a}\|_2^2}{w}$$

Think of [$\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$, …] as a vector.

What does the following quantity represent?

$$\sum_j \boldsymbol{a}_j^2$$

This is the square of the mag

The magnitude of a vector is c
is denoted $\|$

Great exercise: Prove that the L_2 norm of a vector is never greater than the L_1 norm.

$$\|\boldsymbol{a}\|_2 = \sqrt{\sum_j \boldsymbol{a}_j^2}$$

Therefore, our above sum is $\|\boldsymbol{a}\|_2^2$.

$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] \leq \frac{1}{w}\sum_{j \neq i} \boldsymbol{a}_j^2 \leq \frac{\|\boldsymbol{a}\|_2^2}{w}$$

Goal: Make an estimator $\hat{\boldsymbol{a}}$ for some quantity $\boldsymbol{a}$ where

With probability at least $1 - \delta$, — ***Probably***

$|\hat{\boldsymbol{a}} - \boldsymbol{a}| \le \varepsilon \cdot size(input)$ — ***Approximately Correct***

for some measure of the size of the input.

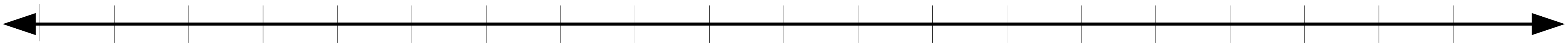
$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] \le \frac{\|\boldsymbol{a}\|_2^2}{w}$$

Goal: Make an estimator $\hat{\boldsymbol{a}}$ for some quantity $\boldsymbol{a}$ where

With probability at least $1 - \delta$, — ***Probably***

$|\hat{\boldsymbol{a}} - \boldsymbol{a}| \leq \varepsilon \cdot \mathit{size}(\mathit{input})$ — ***Approximately Correct***

for some measure of the size of the input.



$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] \leq \frac{\|\boldsymbol{a}\|_2^2}{w}$$

Goal: Make an estimator $\hat{\boldsymbol{a}}$ for some quantity $\boldsymbol{a}$ where

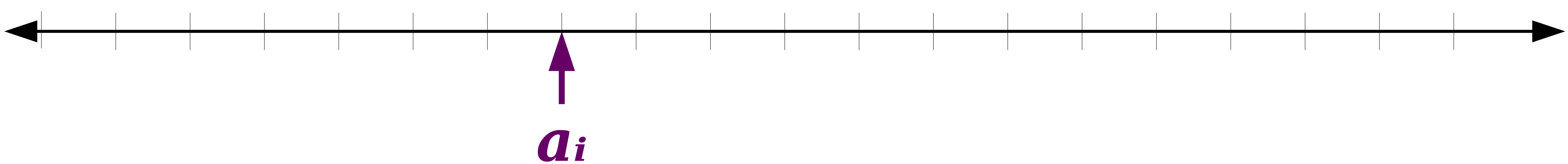
With probability at least 1 – δ,

$|\hat{\boldsymbol{a}} - \boldsymbol{a}| \leq \varepsilon \cdot \mathit{size}(\mathit{input})$

for some measure of the size of the input.

Probably

Approximately Correct



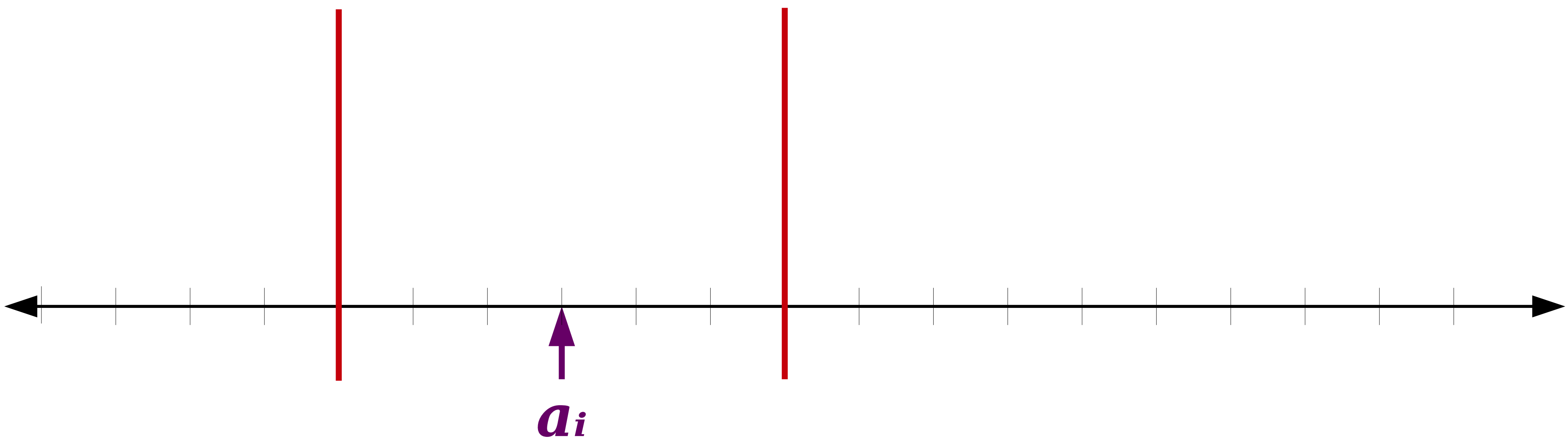
$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] \leq \frac{\|\boldsymbol{a}\|_2^2}{w}$$

Goal: Make an estimator $\hat{\boldsymbol{a}}$ for some quantity $\boldsymbol{a}$ where

With probability at least $1 - \delta$, — ***Probably***

$|\hat{\boldsymbol{a}} - \boldsymbol{a}| \leq \varepsilon \cdot \mathit{size}(\mathit{input})$ — ***Approximately Correct***

for some measure of the size of the input.



$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] \leq \frac{\|\boldsymbol{a}\|_2^2}{w}$$

Goal: Make an estimator $\hat{\boldsymbol{a}}$ for some quantity $\boldsymbol{a}$ where

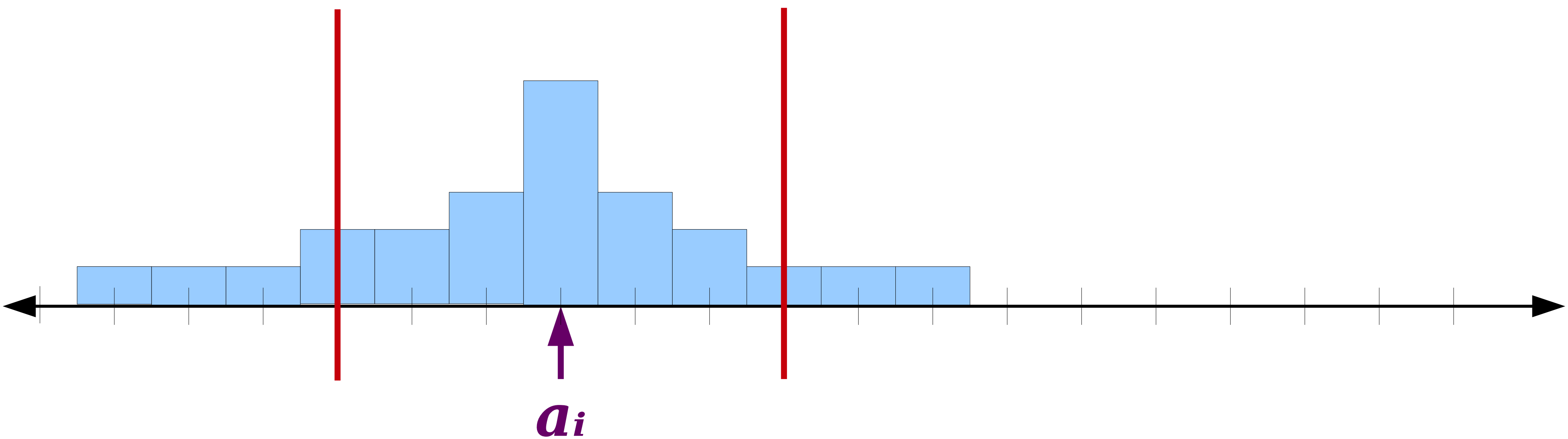
With probability at least $1 - \delta$,

$|\hat{\boldsymbol{a}} - \boldsymbol{a}| \leq \varepsilon \cdot \mathit{size}(\mathit{input})$

for some measure of the size of the input.

Probably

Approximately Correct



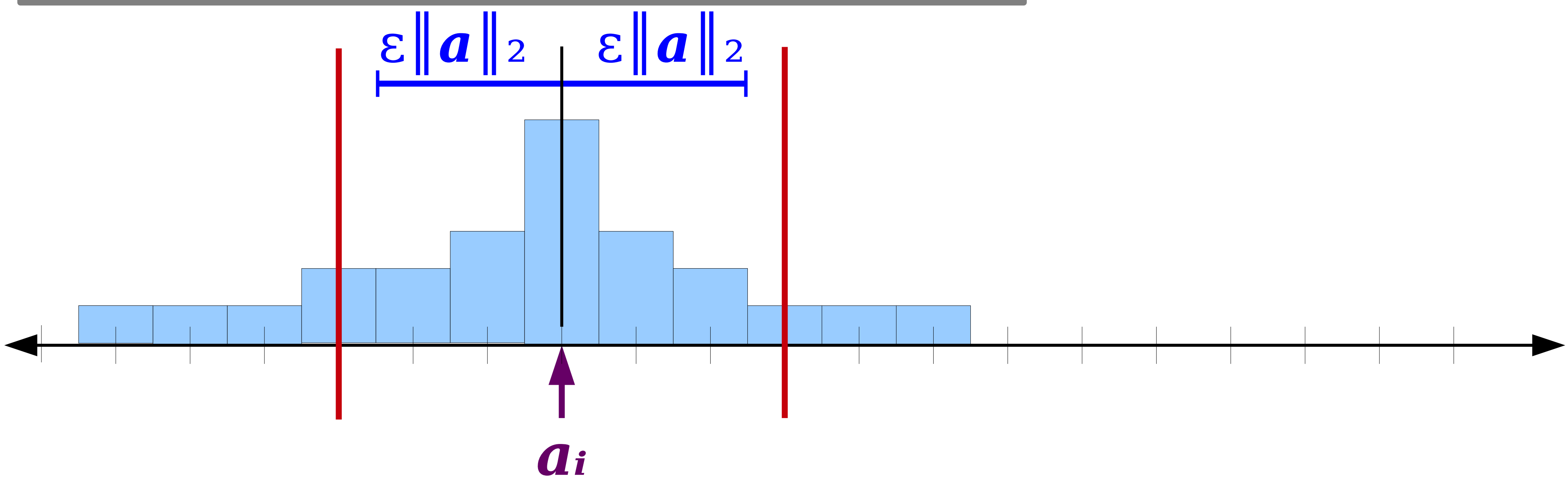
$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] \leq \frac{\|\boldsymbol{a}\|_2^2}{w}$$

Goal: Make an estimator $\hat{\boldsymbol{a}}$ for some quantity $\boldsymbol{a}$ where

With probability at least $1 - \delta$, — ***Probably***

$|\hat{\boldsymbol{a}} - \boldsymbol{a}| \leq \varepsilon \cdot \mathit{size}(\mathit{input})$ — ***Approximately Correct***

for some measure of the size of the input.



$$\mathrm{Var}[\hat{\boldsymbol{a}}_i] \leq \frac{\|\boldsymbol{a}\|_2^2}{w}$$

$$\Pr\left[\left|\hat{\boldsymbol{a}}_i - \boldsymbol{a}_i\right| > \varepsilon \|\boldsymbol{a}\|_2\right]$$

$$\Pr\left[\left|\hat{\boldsymbol{a}}_i - \boldsymbol{a}_i\right| > \varepsilon \|\boldsymbol{a}\|_2\right]$$

Chebyshev's inequality says that

$$\Pr\left[\ \left|X - \mathrm{E}[X]\right| \geq c\ \right] \leq \frac{\mathrm{Var}[X]}{c^2}.$$

$$\Pr\left[\left|\hat{\boldsymbol{a}}_i - \boldsymbol{a}_i\right| > \varepsilon \|\boldsymbol{a}\|_2\right]$$

$$\leq \frac{\mathrm{Var}[\hat{\boldsymbol{a}}_i]}{\left(\varepsilon \|\boldsymbol{a}\|_2\right)^2}$$

Chebyshev's inequality says that

$$\Pr\left[\ \left|X - \mathrm{E}[X]\right| \geq c\ \right] \leq \frac{\mathrm{Var}[X]}{c^2}.$$

$$\Pr\left[\left|\hat{\boldsymbol{a}}_i - \boldsymbol{a}_i\right| > \varepsilon \|\boldsymbol{a}\|_2\right]$$

$$\leq \frac{\mathrm{Var}[\hat{\boldsymbol{a}}_i]}{\left(\varepsilon \|\boldsymbol{a}\|_2\right)^2}$$

$$\boxed{\mathrm{Var}[\hat{\boldsymbol{a}}_i] \leq \frac{\|\boldsymbol{a}\|_2^2}{w}}$$

$$\Pr\left[\left|\hat{\boldsymbol{a}}_i - \boldsymbol{a}_i\right| > \varepsilon\|\boldsymbol{a}\|_2\right]$$

$$\leq \frac{\mathrm{Var}[\hat{\boldsymbol{a}}_i]}{\left(\varepsilon\|\boldsymbol{a}\|_2\right)^2}$$

$$\leq \frac{\|\boldsymbol{a}\|_2^2}{w} \cdot \frac{1}{\left(\varepsilon\|\boldsymbol{a}\|_2\right)^2}$$

$$\boxed{\mathrm{Var}[\hat{\boldsymbol{a}}_i] \leq \frac{\|\boldsymbol{a}\|_2^2}{w}}$$

$$
\begin{aligned}
& \Pr\left[\left|\hat{\boldsymbol{a}}_i - \boldsymbol{a}_i\right| > \varepsilon \|\boldsymbol{a}\|_2\right] \\
\leq \quad & \frac{\mathrm{Var}[\hat{\boldsymbol{a}}_i]}{\left(\varepsilon \|\boldsymbol{a}\|_2\right)^2} \\
\leq \quad & \frac{\|\boldsymbol{a}\|_2^2}{w} \cdot \frac{1}{\left(\varepsilon \|\boldsymbol{a}\|_2\right)^2} \\
= \quad & \frac{1}{w\varepsilon^2}
\end{aligned}
$$

Goal: Make an estimator $\hat{\boldsymbol{a}}$ for some quantity $\boldsymbol{a}$ where

With probability at least $1 - \delta$,

$|\hat{\boldsymbol{a}} - \boldsymbol{a}| \le \varepsilon \cdot \mathit{size}(\mathit{input})$

for some measure of input size.

Probably

Approximately Correct

$$\Pr\left[\left|\hat{\boldsymbol{a}}_i - \boldsymbol{a}_i\right| > \varepsilon \left\|\boldsymbol{a}\right\|_2\right] \le \frac{1}{w\varepsilon^2}$$

We now have a single estimator with a not-so-great chance of giving a good estimate.

How do we fix this?

Pick $w = 4 \cdot \varepsilon^{-2}$. Then

$$\Pr\left[\left|\hat{\boldsymbol{a}}_i - \boldsymbol{a}_i\right| > \varepsilon \left\|\boldsymbol{a}\right\|_2\right] \le \frac{1}{4}.$$

How to Build an Estimator

	Count-Min Sketch	***Count Sketch***
Step One: Build a Simple Estimator	Hash items to counters; add +1 when item seen.	Hash items to counters; add ±1 when item seen.
Step Two: Compute Expected Value of Estimator	Sum of indicators; 2-independent hashes have low collision rate.	2-independence breaks up products; ±1 variables have zero expected value.
Step Three: Apply Concentration Inequality	One-sided error; use expected value and Markov's inequality.	Two-sided error; compute variance and use Chebyshev's inequality.
Step Four: Replicate to Boost Confidence	Take min; only fails if all estimates are bad.	

How to Build an Estimator

	Count-Min Sketch	***Count Sketch***
Step One: Build a Simple Estimator	Hash items to counters; add +1 when item seen.	Hash items to counters; add ±1 when item seen.
Step Two: Compute Expected Value of Estimator	Sum of indicators; 2-independent hashes have low collision rate.	2-independence breaks up products; ±1 variables have zero expected value.
Step Three: Apply Concentration Inequality	One-sided error; use expected value and Markov's inequality.	Two-sided error; compute variance and use Chebyshev's inequality.
Step Four: Replicate to Boost Confidence	Take min; only fails if all estimates are bad.	

Running in Parallel

- Imagine we call ***estimate***(x) on each of our estimators and get back these estimates.
- We need to give back a single number.
- ***Question:*** How should we aggregate these numbers into a single estimate?

Formulate a hypothesis, but ***don't post anything in chat just yet***.

Estimator 1: 137	*Estimator 2:* 271	*Estimator 3:* 166	*Estimator 4:* 103	*Estimator 5:* 261

Running in Parallel

- Imagine we call ***estimate***(x) on each of our estimators and get back these estimates.
- We need to give back a single number.
- ***Question:*** How should we aggregate these numbers into a single estimate?

Now, ***private chat me your best guess***. Not sure? Just answer "??".

Estimator 1: 137	*Estimator 2:* 271	*Estimator 3:* 166	*Estimator 4:* 103	*Estimator 5:* 261

Running in Parallel

- Imagine we call ***estimate***(x) on each of our estimators and get back these estimates.
- We need to give back a single number.
- ***Question:*** How should we aggregate these numbers into a single estimate?

Estimator 1: 137	*Estimator 2:* 271	*Estimator 3:* 166	*Estimator 4:* 103	*Estimator 5:* 261

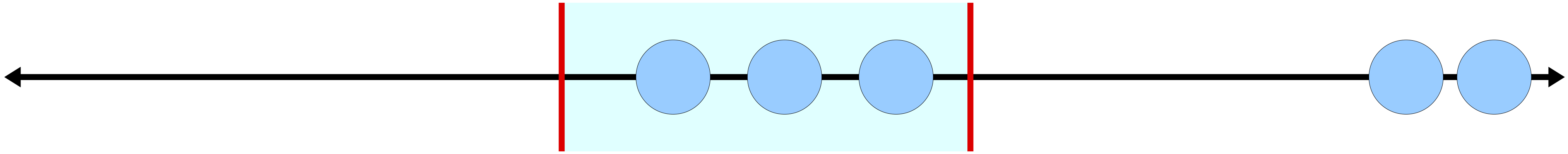
Running in Parallel

- Unlike last time, we have a two-sided error, so taking the minimum would be a Very Bad Thing.
- Two reasonable options come to mind:
  - Take the ***mean*** of the estimates.
  - Take the ***median*** of the estimates.
- ***Question:*** Which should we pick?

Estimator 1:	*Estimator 2:*	*Estimator 3:*	*Estimator 4:*	*Estimator 5:*
137	271	166	103	261

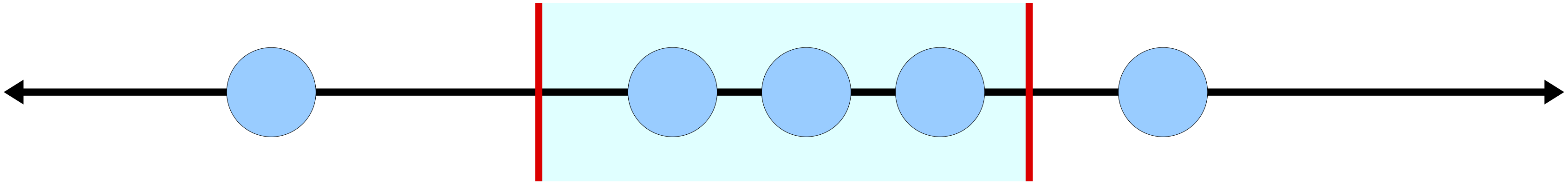
Working With Means

- ***Claim:*** Taking the mean of multiple estimators does increase our probability of being close to the expected value, but not very quickly.
- ***Intuition:*** Not all outliers are created equal, and outliers far from the target range skew the estimate.
- ***The Math:*** Averaging d copies of an estimator decreases the variance by a factor of d. (Prove this!) By Chebyshev, that decreases the probability of getting a bad answer by a factor of d. We'd like something that decays exponentially in d.



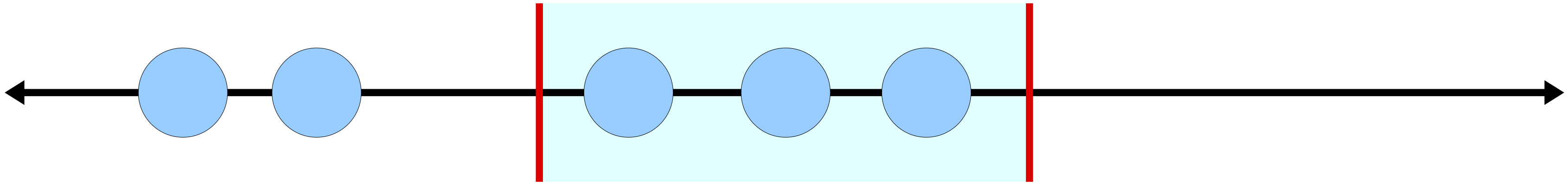
Working With Medians

- ***Claim:*** If we output the median estimate given by the data structures, we have high probability of giving an acceptably close answer.
- ***Intuition:*** The only way that the median isn't in the "good" area is if ***at least half*** the estimates are in the "bad" area.
- Each individual data structure has a "reasonable" chance to be good, so this is very unlikely.



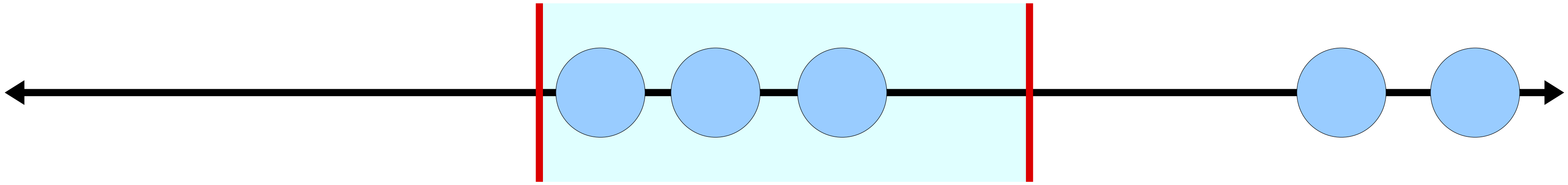
Working With Medians

- ***Claim:*** If we output the median estimate given by the data structures, we have high probability of giving an acceptably close answer.
- ***Intuition:*** The only way that the median isn't in the "good" area is if ***at least half*** the estimates are in the "bad" area.
- Each individual data structure has a "reasonable" chance to be good, so this is very unlikely.



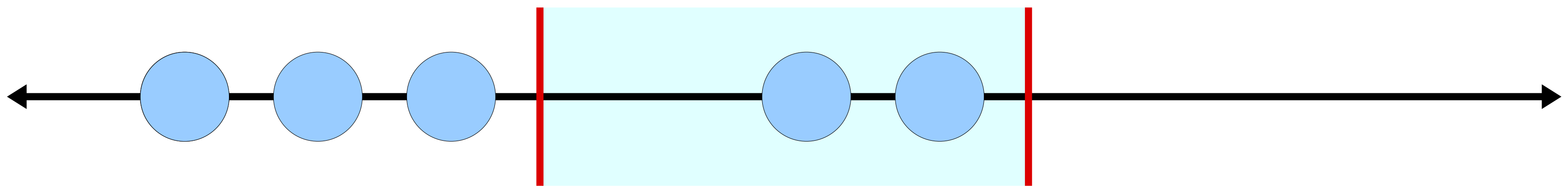
Working With Medians

- ***Claim:*** If we output the median estimate given by the data structures, we have high probability of giving an acceptably close answer.
- ***Intuition:*** The only way that the median isn't in the "good" area is if ***at least half*** the estimates are in the "bad" area.
- Each individual data structure has a "reasonable" chance to be good, so this is very unlikely.



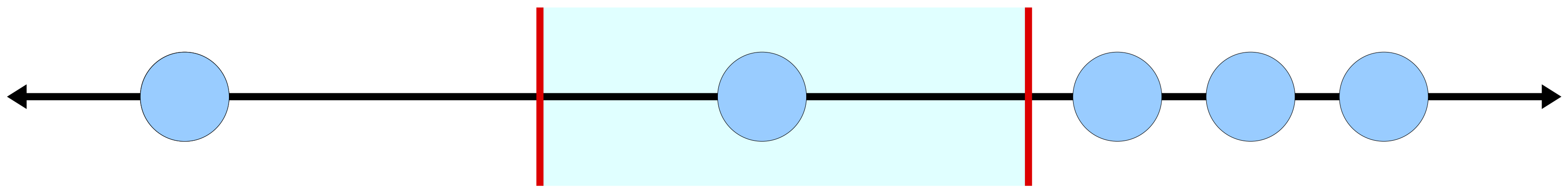
Working With Medians

- ***Claim:*** If we output the median estimate given by the data structures, we have high probability of giving an acceptably close answer.
- ***Intuition:*** The only way that the median isn't in the "good" area is if ***at least half*** the estimates are in the "bad" area.
- Each individual data structure has a "reasonable" chance to be good, so this is very unlikely.



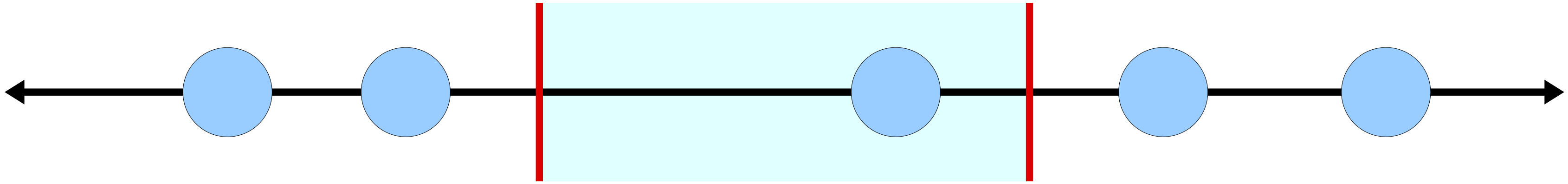
Working With Medians

- ***Claim:*** If we output the median estimate given by the data structures, we have high probability of giving an acceptably close answer.
- ***Intuition:*** The only way that the median isn't in the "good" area is if ***at least half*** the estimates are in the "bad" area.
- Each individual data structure has a "reasonable" chance to be good, so this is very unlikely.



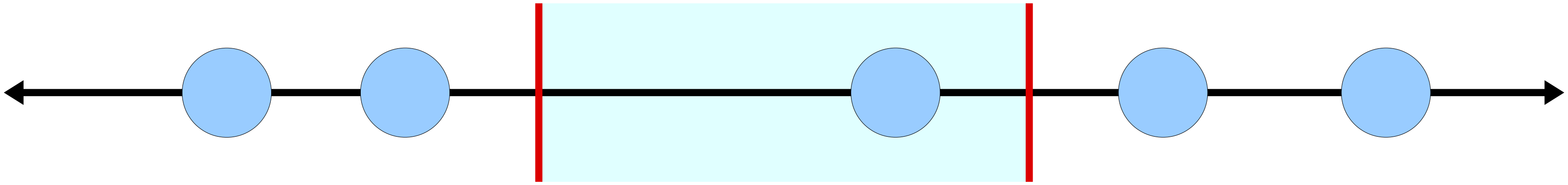
Working With Medians

- ***Claim:*** If we output the median estimate given by the data structures, we have high probability of giving an acceptably close answer.
- ***Intuition:*** The only way that the median isn't in the "good" area is if ***at least half*** the estimates are in the "bad" area.
- Each individual data structure has a "reasonable" chance to be good, so this is very unlikely.



Working With Medians

- Let D denote a random variable equal to the number of data structures that produce an answer *not* within $\varepsilon||\boldsymbol{a}||_2$ of the true answer.
- Since each independent data structure has failure probability at most ¼, we can upper-bound D with a Binom(d, ¼) variable.
- We want to know $\Pr[D > d / 2]$.
- How can we determine this?



Chernoff Bounds

- The ***Chernoff bound*** says that if $X \sim \text{Binom}(n, p)$ and $p < ½$, then

  $$\Pr\left[X \geq \frac{n}{2}\right] < e^{-n \cdot z(p)}$$

  where $z(p) = (½ - p)^2 / 2p$.

Chernoff Bounds

- The ***Chernoff bound*** says that if $X \sim \text{Binom}(n, p)$ and $p < ½$, then

$$\Pr[X \geq \frac{n}{2}] < e^{-n \cdot z(p)}$$

where $z(p) = (½ - p)^2 / 2p$.

Intuition: For any fixed value of p, this quantity decays exponentially quickly as a function of n. It's extremely unlikely that more than half our estimates will be bad.

Chernoff Bounds

- The ***Chernoff bound*** says that if $X \sim \text{Binom}(n, p)$ and $p < ½$, then

$$\Pr[X \geq \frac{n}{2}] < e^{-n \cdot z(p)}$$

  where $z(p) = (½ - p)^2 / 2p$.

- In our case, $D \sim \text{Binom}(d, ¼)$, so we know that

$$\Pr[D \geq \frac{d}{2}] \leq e^{-n \cdot z(1/4)} = e^{-d/8}$$

- Therefore, choosing $\boldsymbol{d = 8 \ln \delta^{-1}}$ ensures that

$$\Pr[|\hat{\boldsymbol{a}}_i - \boldsymbol{a}_i| > \varepsilon \|\boldsymbol{a}\|_2] \leq \Pr[D \geq \frac{d}{2}] \leq \delta$$

How to Build an Estimator

	Count-Min Sketch	***Count Sketch***
Step One: Build a Simple Estimator	Hash items to counters; add +1 when item seen.	Hash items to counters; add ±1 when item seen.
Step Two: Compute Expected Value of Estimator	Sum of indicators; 2-independent hashes have low collision rate.	2-independence breaks up products; ±1 variables have zero expected value.
Step Three: Apply Concentration Inequality	One-sided error; use expected value and Markov's inequality.	Two-sided error; compute variance and use Chebyshev's inequality.
Step Four: Replicate to Boost Confidence	Take min; only fails if all estimates are bad.	Take median; only can fail if half of estimates are wrong; use Chernoff.

The Count Sketch

$w = \lceil 4 \cdot \varepsilon^{-2} \rceil$

h_1	s_1
h_2	s_2
h_3	s_3
…	…
h_d	s_d

31	41	-59	-26	…	58
27	-18	28	-18	…	-45
16	-18	-3	39	…	-75
…					
69	-31	47	-18	…	59

$d = \lceil 8 \ln \delta^{-1} \rceil$

Sampled uniformly and independently from 2-independent families of hash functions

The Count Sketch

$w = \lceil 4 \cdot \varepsilon^{-2} \rceil$

h_1	s_1
h_2	s_2
h_3	s_3
…	…
h_d	s_d

31	41	-59	-26	…	58
27	-18	28	-18	…	-45
16	-18	-3	39	…	-75
…					
69	-31	47	-18	…	59

$d = \lceil 8 \ln \delta^{-1} \rceil$

```
increment(x):
   for i = 1 … d:
      count[i][hi(x)] += si(x)
```

The Count Sketch

$w = \lceil 4 \cdot \varepsilon^{-2} \rceil$

$d = \lceil 8 \ln \delta^{-1} \rceil$

h_1	s_1
h_2	s_2
h_3	s_3
…	…
h_d	s_d

31	**41**	-59	-26	…	58
27	-18	28	**-18**	…	-45
16	-18	-3	**39**	…	-75
		…			
69	-31	47	-18	…	**59**

```
increment(x):
   for i = 1 … d:
      count[i][hᵢ(x)] += sᵢ(x)
```

The Count Sketch

$w = \lceil 4 \cdot \varepsilon^{-2} \rceil$

h_1	s_1	31	40	-59	-26	…	58
h_2	s_2	27	-18	28	-19	…	-45
h_3	s_3	16	-18	-3	40	…	-75
…	…			…			
h_d	s_d	69	-31	47	-18	…	58

$d = \lceil 8 \ln \delta^{-1} \rceil$

```
increment(x):
   for i = 1 … d:
      count[i][hi(x)] += si(x)
```

The Count Sketch

$w = \lceil 4 \cdot \varepsilon^{-2} \rceil$

h_1	s_1
h_2	s_2
h_3	s_3
…	…
h_d	s_d

31	40	-59	-26	…	58
27	-18	28	-19	…	-45
16	-18	-3	40	…	-75
…					
69	-31	47	-18	…	58

$d = \lceil 8 \ln \delta^{-1} \rceil$

```
increment(x):
   for i = 1 … d:
      count[i][hi(x)] += si(x)
```

The Count Sketch

$w = \lceil 4 \cdot \varepsilon^{-2} \rceil$

h_1	s_1
h_2	s_2
h_3	s_3
…	…
h_d	s_d

31	40	-59	-26	…	58
27	-18	28	-19	…	-45
16	-18	-3	40	…	-75
		…			
69	-31	47	-18	…	58

$d = \lceil 8 \ln \delta^{-1} \rceil$

```
increment(x):
   for i = 1 … d:
      count[i][hi(x)] += si(x)
```

```
estimate(x):
   options = []
   for i = 1 … d:
      options += count[i][hi(x)] * si(x)
   return medianOf(options)
```

The Count Sketch

$w = \lceil 4 \cdot \varepsilon^{-2} \rceil$

$d = \lceil 8 \ln \delta^{-1} \rceil$

h_1	s_1
h_2	s_2
h_3	s_3
…	…
h_d	s_d

31	40	-59	-26	…	58
27	-18	28	-19	…	-45
16	-18	-3	40	…	-75
…					
69	-31	47	-18	…	58

```
increment(x):
   for i = 1 … d:
      count[i][hi(x)] += si(x)
```

```
estimate(x):
   options = []
   for i = 1 … d:
      options += count[i][hi(x)] * si(x)
   return medianOf(options)
```

The Final Analysis

- Here's a comparison of these two structures.
- ***Question to ponder:*** When is a count-min sketch better than a count sketch, and vice-versa?

Count-Min Sketch

Space: $\Theta(\varepsilon^{-1} \cdot \log \delta^{-1})$

increment: $\Theta(\log \delta^{-1})$

estimate: $\Theta(\log \delta^{-1})$

Accuracy: within $\varepsilon||\boldsymbol{a}||_1$.

Count Sketch

Space: $\Theta(\varepsilon^{-2} \cdot \log \delta^{-1})$

increment: $\Theta(\log \delta^{-1})$

estimate: $\Theta(\log \delta^{-1})$

Accuracy: within $\varepsilon||\boldsymbol{a}||_2$

Major Ideas Here

- Concentration inequalities are useful tools for showing the right thing probably happens.
  - For one-sided errors, try Markov's inequality.
  - For two-sided errors, try Chebyshev's inequality.
  - To bound the probability that lots of things all go wrong, use Chernoff bounds.
  - For more on different mathematical tools like these, check out this blog post by Scott Aaronson.
- Modest success probability can be amplified by running things in parallel.
  - For one-sided errors, try using the min or max.
  - For two-sided errors, try using the median.
- We can estimate quantities using significantly less space than storing those quantities exactly if we're okay with approximate answers.

Cardinality Estimation

Cardinality Estimation

- A ***cardinality estimator*** is a data structure supporting the following operations:
  - ***see***(x), which records that x has been seen, and
  - ***estimate***(), which returns an estimate of the number of ***distinct*** values we've seen.
- In other words, they estimate the cardinality of the set of all items that have been seen.
- These data structures are widely deployed in practice.
  - Databases use them to select which of many different algorithms to run, based on the number of items to process.
  - Websites use them to estimate how many different people have visited the site in a given time window.

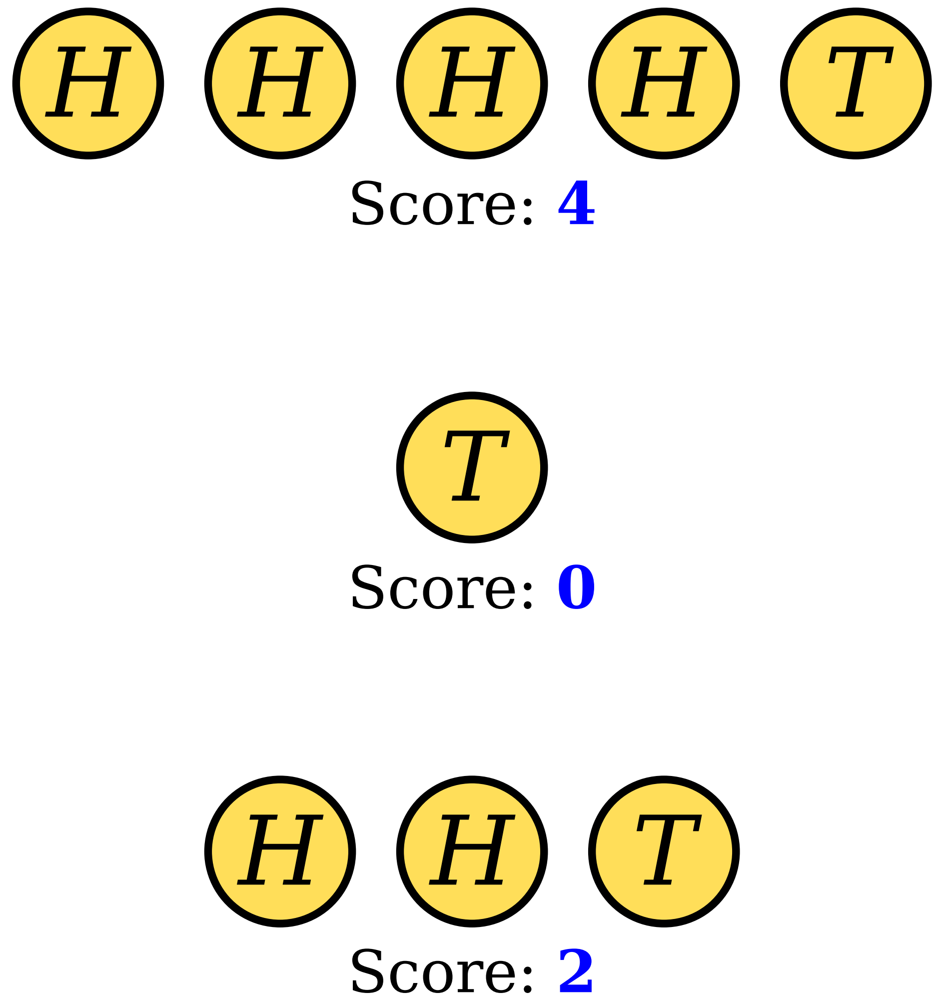
Cardinality Estimation

- As with frequency estimation, we can solve the cardinality estimation problem exactly using hash tables or binary search trees using $\Omega(n)$ space.
- To be useful in large-scale data applications, cardinality estimators need to use *significantly* less space than this.
- ***Question:*** How low can we go?

Flipping Coins

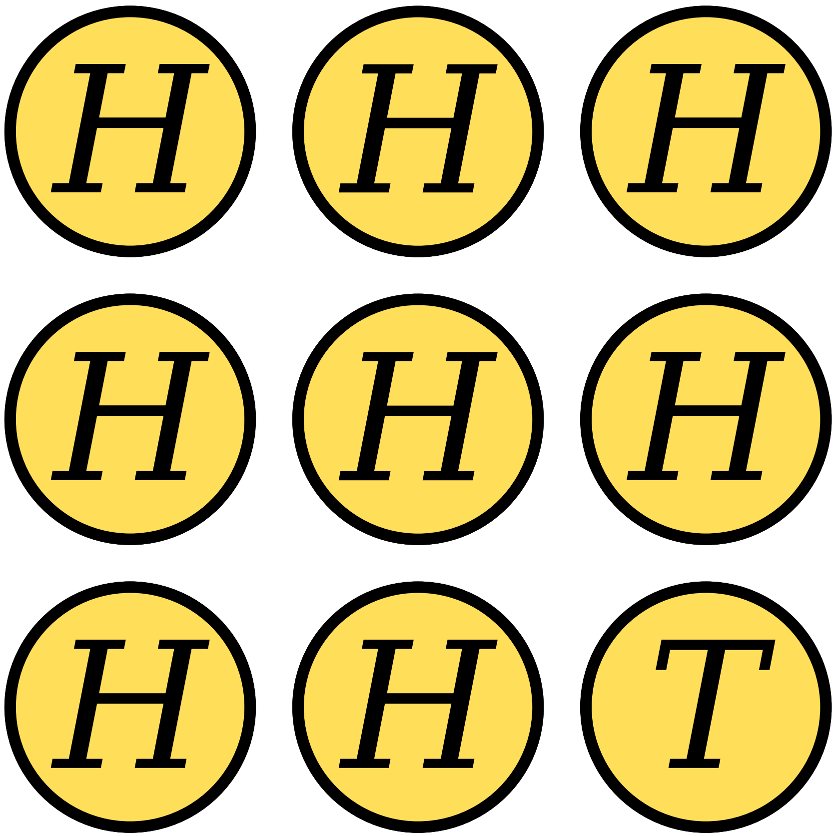
Flipping Coins

- Here's a game: I'm going to flip a coin until I get tails. My score is the number of heads that I flip.
- The probability of flipping k or more consecutive heads is 2^{-k}, so it's pretty unlikely that I'm going to flip lots of heads in a row.



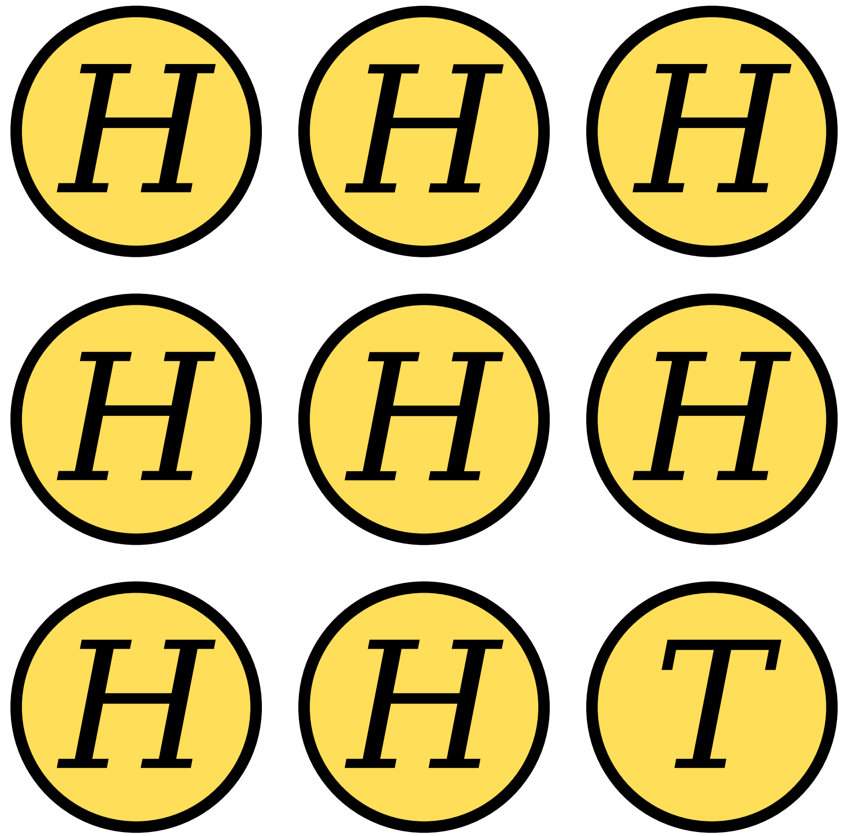
Flipping Coins

- Suppose I show you the following clip of me playing this game.
- Which is more likely?
  - I played the game once and got really lucky.
  - I played the game 256 times and showed you my best run.
- Probability you see this after one game: $^1/_{512}$.
- Probability this is the best you see after 256 games: approximately 23.3%.



Flipping Coins

- ***Intuition:*** Play this game multiple times and track the maximum number of heads you get in a row.
- If the maximum number of heads we see is H, estimate that we played 2^H times.
- ***Question:*** How good of an estimate is this?

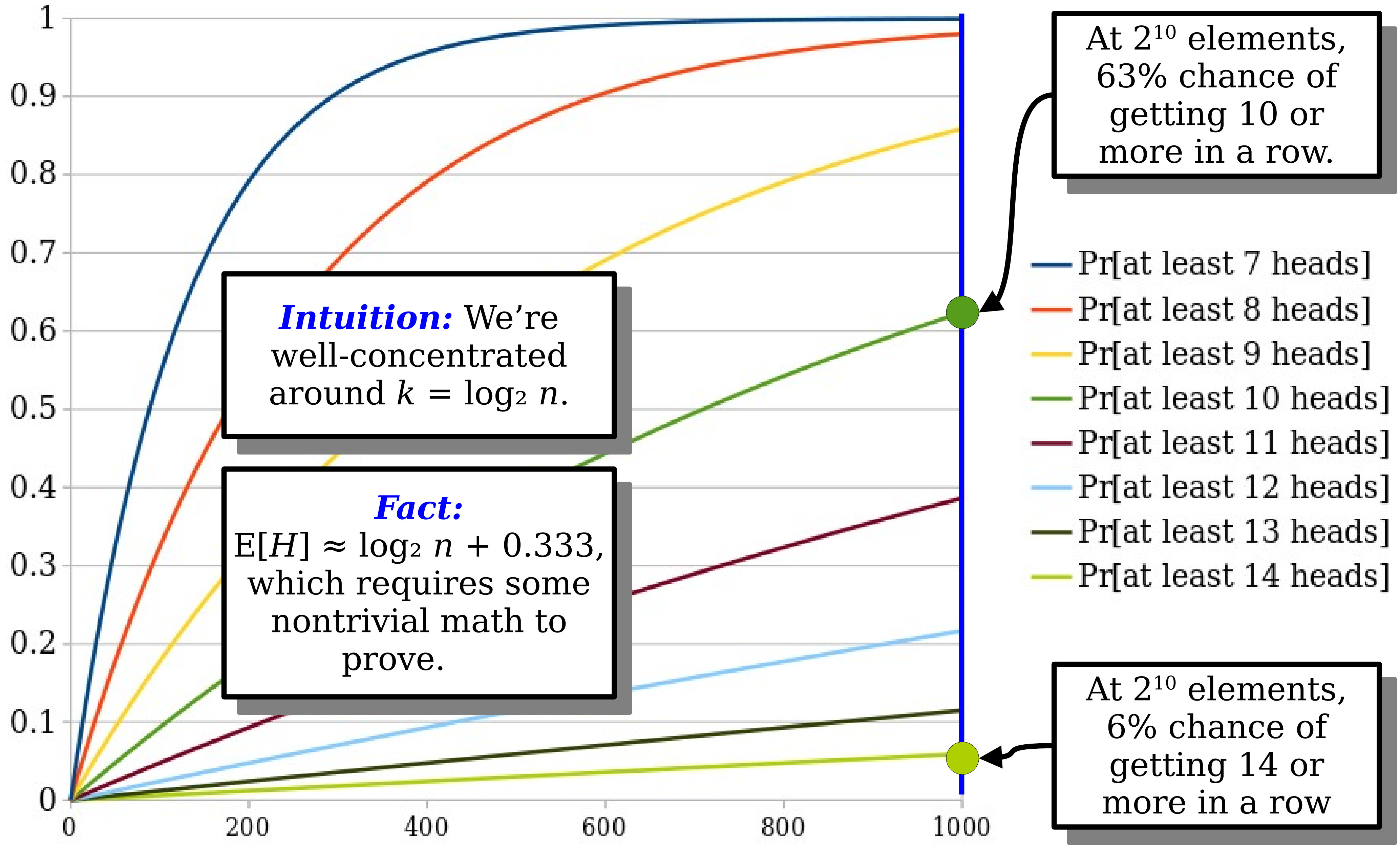


Flipping Coins

- Suppose we play this game n times. What's the probability we see at least k consecutive heads at least once?

$$
\begin{aligned}
&\Pr[\text{see at least } k \text{ heads in } n \text{ games}] \\
&= 1 - \Pr[\text{never see } k \text{ heads in } n \text{ games}] \\
&= 1 - \Pr[\text{never see } k \text{ heads in one game}]^n \\
&= 1 - (1 - 2^{-k})^n
\end{aligned}
$$

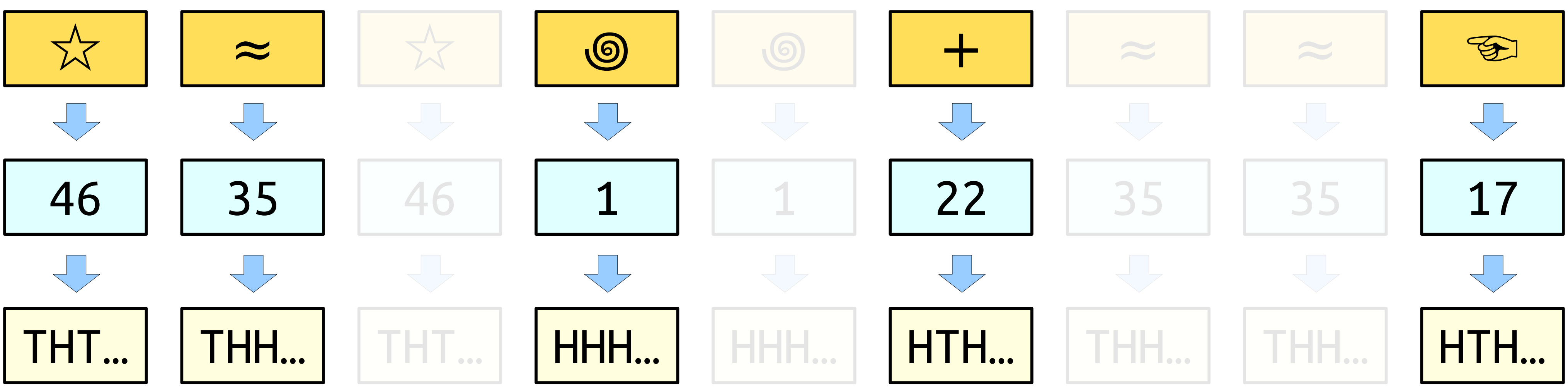
- What does this function look like?



Play this game *n* times. What is the probability that our maximum score is *k* or more?

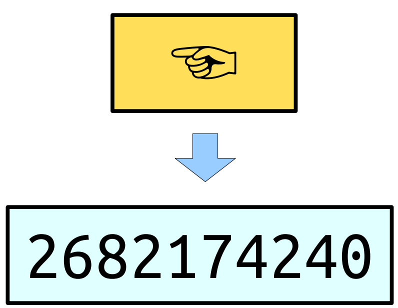
From Coins to Cardinality

- Ultimately, we're interested in building a cardinality estimator. How does this help us?
- ***Idea:*** Hash each item in the data stream, and use each hash as the random source for the coin-flipping game.
- Duplicate items give duplicate hashes, which provide duplicate games, which function as if they never happened.



From Coins to Cardinality

- We need some way of going from hash codes to sequences of coin tosses.
- ***Idea:*** Treat the hash as a sequence of bits. 0 means heads, 1 means tails.



From Coins to Cardinality

- We need some way of going from hash codes to sequences of coin tosses.
- ***Idea:*** Treat the hash as a sequence of bits. 0 means heads, 1 means tails.
- Then, count how many 0 bits appear consecutively at the end of the number.

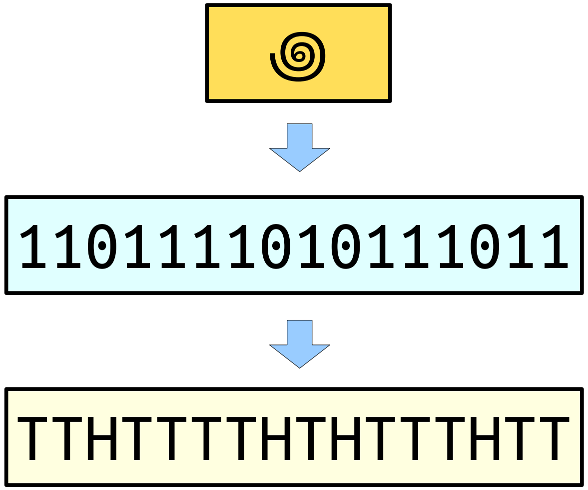
☜

10011111110111101011101100100000

THHTTTTTTTHTTTTHTHTTTHTTHHTHHHHH

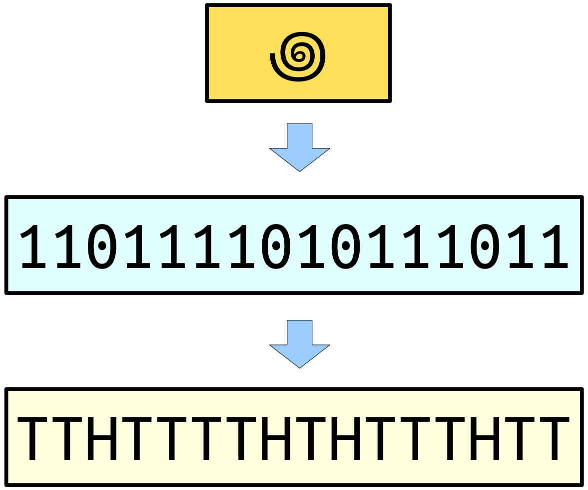
A Simple Estimator

- Keep track of a value H, initially zero, that records the maximum number of zero bits seen at the end of a number.
- To ***see*** an item:
  - Compute a hash code for that item.
  - Compute the number of trailing zeros.
  - Update H if this is a new record.
- To ***estimate*** the number of distinct elements:
  - Return 2^H.



A Simple Estimator

- How much space does this single estimator need?
- Assume we have an upper bound U on the maximum cardinality. Our hashes never need more than $\Theta(\log U)$ bits.
- Bits required to write down the position of a bit in that hash: $\Theta(\log \log U)$.
- That is an *absolutely tiny* amount of space compared to storing the elements!



Improving the Estimator

- The current estimator has a few weaknesses.
  - It always outputs a size that's a power of two, so we're likely to be off by a full binary order of magnitude.
  - It tends to skew high, since a single unexpected run of heads pushes the whole total up.
- But we have already seen some techniques for improving estimators:
  - Run lots of copies in parallel to reduce the likelihood of any one of them being bad.
  - Use some creative strategy to combine those individual estimates into one really good one.
- And in fact, folks have done just that.

HyperLogLog

- The ***HyperLogLog*** estimator uses many independent copies of this estimator to produce a very high-quality estimate.
  - Run *m* copies of the estimator, using a hash function to distribute items to estimators, so that each copy gets roughly a $1/m$ fraction of the items.
  - Compute the *harmonic mean* of the estimates to mitigate outliers while smoothing between powers of two.
  - Multiply in a debiasing term to mitigate the skew from both the original estimates and the harmonic mean.
- This estimator is used extensively in practice; with about 768 bytes of memory, it can estimate cardinalities for any real-world data stream to about 3% accuracy.
- It's widely used in database systems, and many open-source implementations are available.

HyperLogLog

- The analysis of HyperLogLog from the original paper is exceedingly difficult, and I haven't been able to follow along with all the details.
- Hopefully, this intuitive explanation of how it works is enough for you.
- ***(Probably?) Open problem:*** Find a significantly simpler and cleaner rigorous analysis of HyperLogLog than the original.

Major Ideas We’ve Seen

- You can build a great estimator by running lots of weak estimators in parallel and aggregating the results.
- Indicator variables and linearity of expectation are powerful tools when analyzing sketches.
- Markov’s and Chebyshev’s inequalities are useful for bounding probabilities involving hashing.
- The Chernoff bound is a great tool for showing it’s unlikely for lots of things to go wrong.

Next Time

- ***Computational Geometry***
  - Data structures for points in space.
- ***Orthogonal Range Searching***
  - Finding all points in a box.
- ***Geometric Cascading***
  - Saving time when doing binary searches.