Cardinality

Ben Langmead

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Department of Computer Science

Cardinality

3201
946
5581
8945
6145
8126
3887
8925
1246
8324
4549
9100
5598
8499
8970
3921
8575
4859
4960
42
6901
4336
9228
3317
399

Cardinality: how many *distinct* values in a data stream?

Cardinality

3201
946
5581
8945
6145
8126
3887
8925
1246
8324
4549
9100
5598
8499
8970
3921
8575
4859
4960
42
6901
4336
9228
3317
399

Strategies:

Quadratic scan, count

$O(m^2)$ time for scan
$O(m)$ space for items

Sort, linear scan, count

$O(m \log m)$ time for sort
$O(m)$ space for items

Cardinality

3201
946
5581
8945
6145
8126
3887
8925
1246
8324
4549
9100
5598
8499
8970
3921
8575
4859
4960
42
6901
4336
9228
3317
399

Strategies:

Direct-address the universe

Every universe item gets a slot in a bitvector initialized to all 0s

For each item, set corresponding bitvector item to 1. Count flips as we go.

$O(|U|)$ bits of space

Cardinality

3201
946
5581
8945
6145
8126
3887
8925
1246
8324
4549
9100
5598
8499
8970
3921
8575
4859
4960
42
6901
4336
9228
3317
399

Strategies:

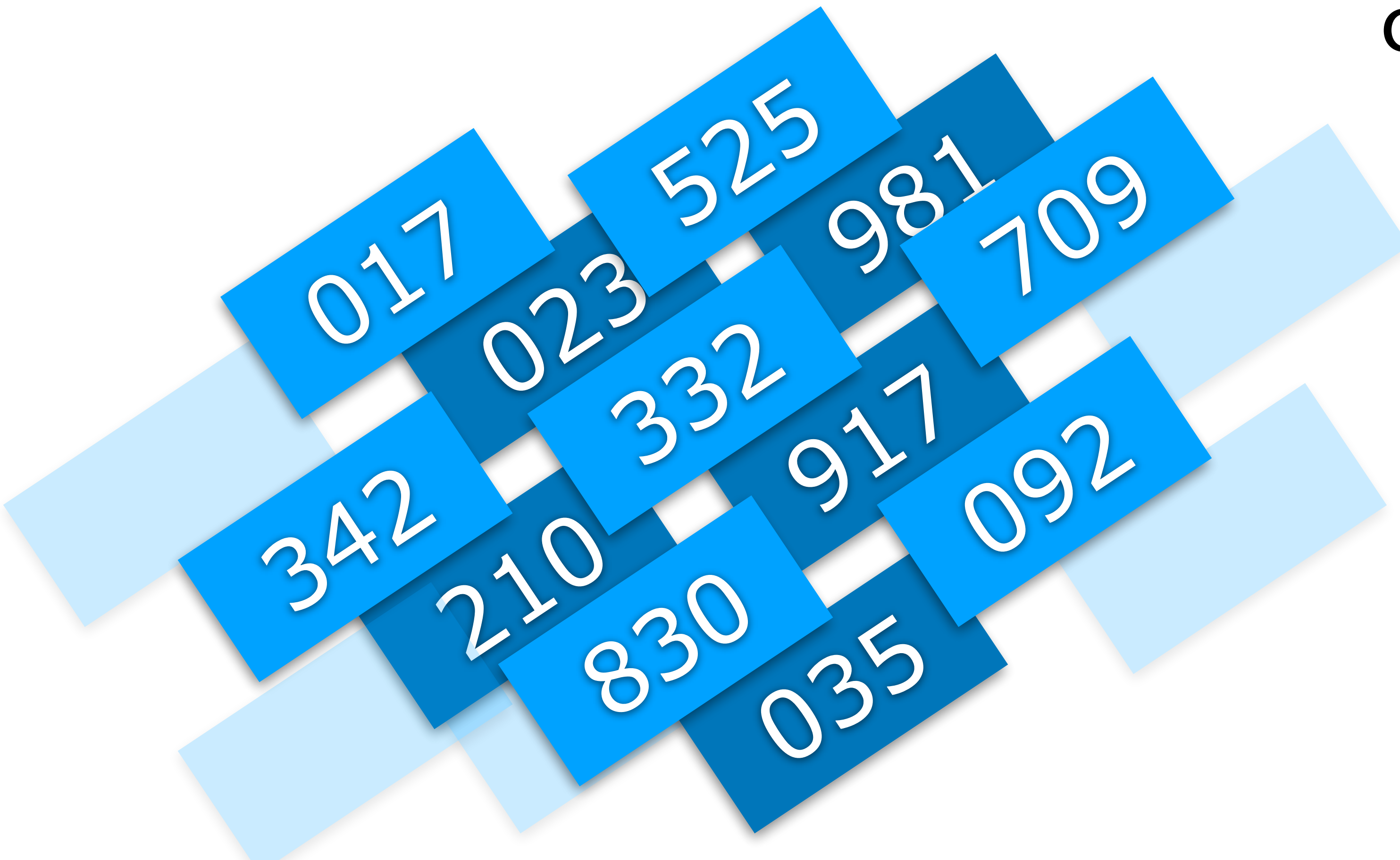
Hash table

$O(|\text{distinct}(M)|)$ space

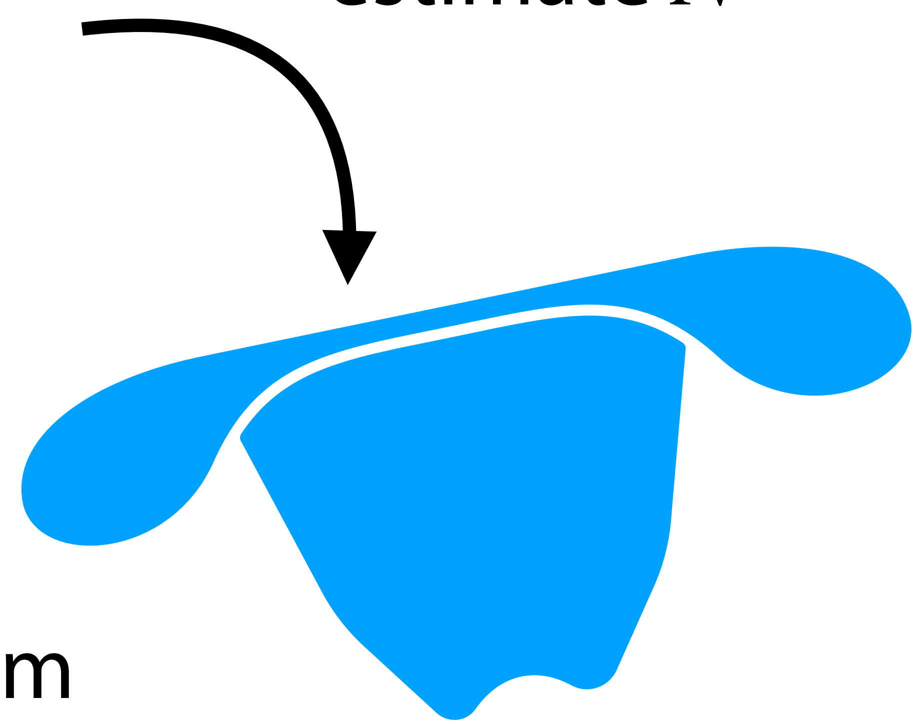
(If approximating, can use Bloom filter!)

Cardinality

I take cards labeled 1--1,000 and choose a random subset of size N to hide in my hat



You would like to estimate N



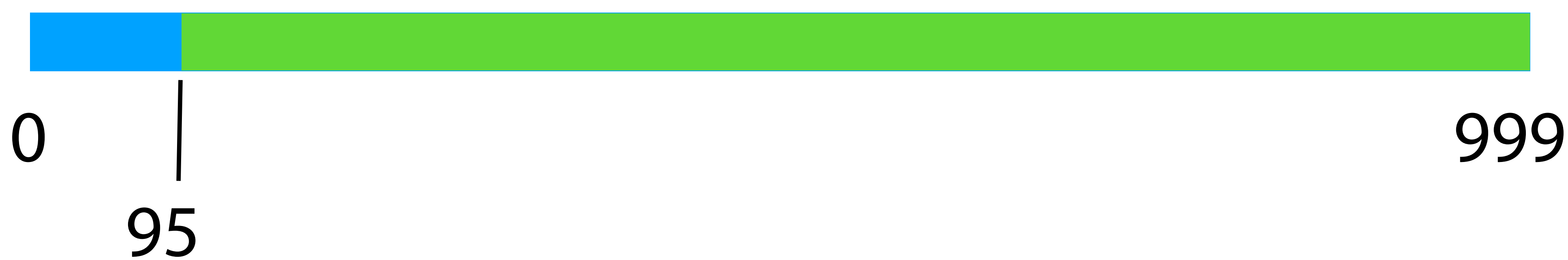
You may see **one representative** from the cards in the hat; which to pick?

Minimum, median, maximum? Something else?

Cardinality

What if **minimum** was 500? ...10? ... 4?

If minimum is 95, what's our estimate for N?



Informally: N points scattered randomly across interval divide it in $N+1$ parts, each about $1000/(N+1)$ long

$$95 \approx 1000/(N+1)$$
$$N+1 \approx 10.5$$
$$N \approx 9.5$$

Minimum is very easy to calculate

Cardinality

Sampling

Choose representatives **randomly**

Sample statistics shed light on population statistics

Sketching

Choose representatives **deterministically**

Composable; unions are natural

Can be designed not to miss extreme / informative items

Minimum is a ***deterministic*** choice made when sketching. Contrast with what we wold have learned from making a random choice.

Cardinality

With *minimum*, it doesn't matter whether input items are repeated

...contrast with sampling

91
38
46
75
82
59
78
72
98
27
77
33
86
82
2
47
31
17
69
77
18
3
22
2
54

Cardinality

Let $M = \min(X_1, X_2, \ldots, X_N)$, where each X_i is an independent uniform draw from the reals in [0, 1]

X_is model the hash values for the N items

Claim: $\mathbf{E}[M] = \dfrac{1}{N+1}$

Cardinality

A hash with say 32-bit or 64-bit output can be thought of as outputting reals in [0, 1]

Say h_{64} is a 64-bit hash function; $\dfrac{h_{64}(x)}{2^{64}-1}$ spreads outputs along [0, 1] with super fine resolution

Cardinality

Draws:	X_1	X_2	X_3	...	X_N
Min indicators:	I_1	I_2	I_3	...	I_N

$$M = \min_{1 \le i \le N} X_i$$

$$I_i = \begin{cases} 1 & \text{if } \; X_i < \min_{j \neq i} X_j \\ 0 & \text{otherwise} \end{cases}$$

Cardinality

Scenario 1

X_i	0.455	0.220	0.951	0.236	0.979
I_i	0	1	0	0	0

$$M = 0.220$$

Scenario 2

X_i	0.968	0.234	0.835	0.642	0.349
I_i	0	1	0	0	0

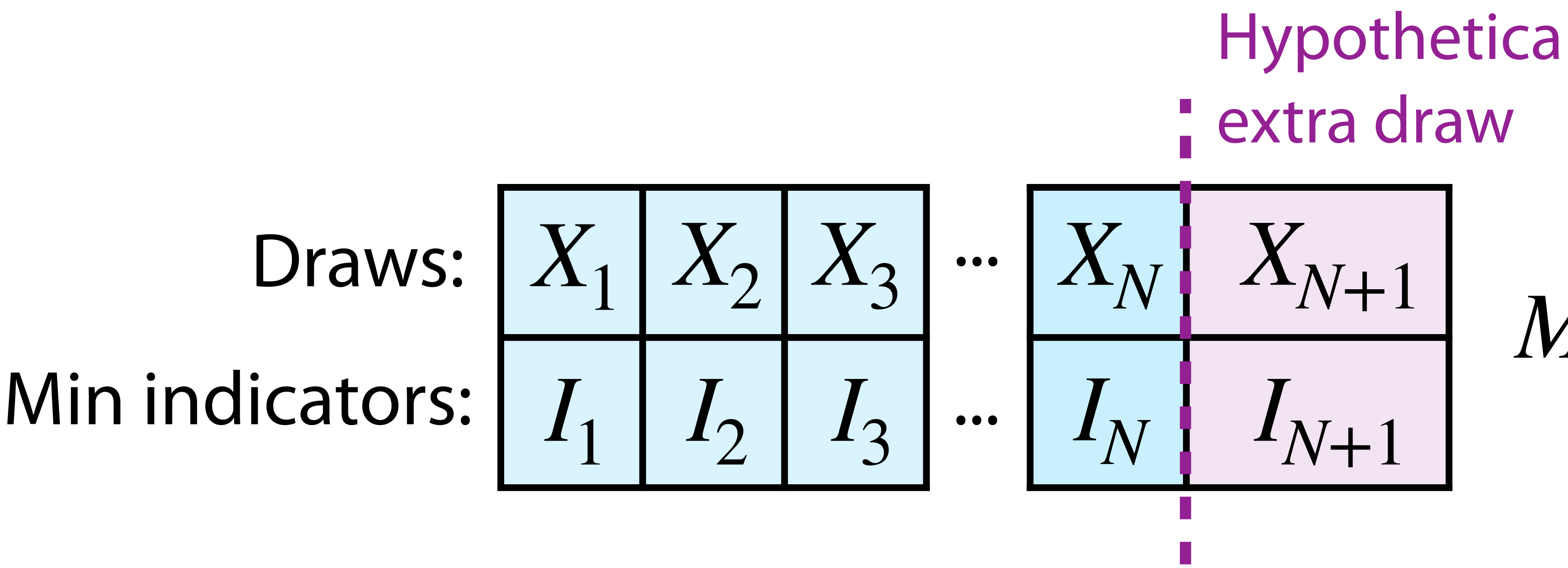
$$M = 0.234$$

Scenario 3

X_i	0.774	0.484	0.309	0.526	0.143
I_i	0	0	0	0	1

$$M = 0.143$$

Cardinality



$$M = \min_{1 \leq i \leq N} X_i$$

$$I_i = \begin{cases} 1 & \text{if } \; X_i < \min_{j \neq i} X_j \\ 0 & \text{otherwise} \end{cases}$$

Cardinality

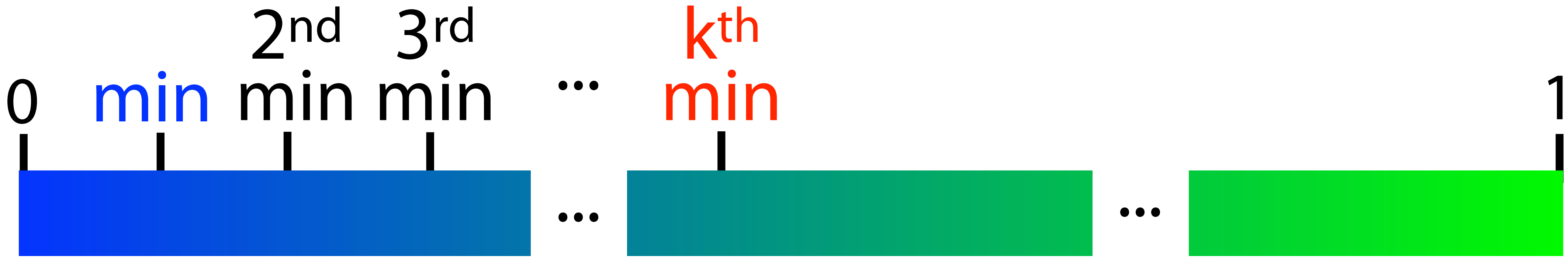
By symmetry, for each i, $\mathbf{E}[I_i] = \dfrac{1}{N+1}$

(Draws have equal chance of being minimum)

$$\frac{1}{N+1} = \mathbf{E}[I_{N+1}] = \Pr\left(X_{N+1} < \min_{1 \le i \le N} X_i\right) = \mathbf{E}[M]$$

Cardinality

Can the k^{th}-smallest hash value estimate the cardinality better than the minimum?



Cardinality

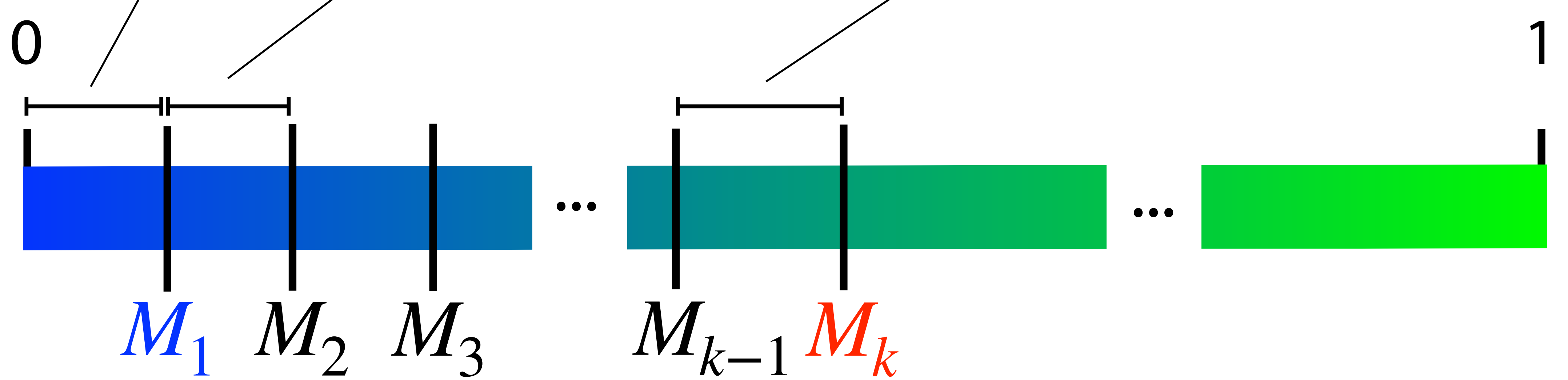
Can the k^{th}-smallest hash value estimate the cardinality better than the minimum?

$$\mathbf{E}[M_1] = \frac{1}{N+1} \qquad \mathbf{E}[M_k] = \frac{k}{N+1}$$

Cardinality

$$\frac{1}{N+1} = \frac{\mathbf{E}[M_k]}{k}$$

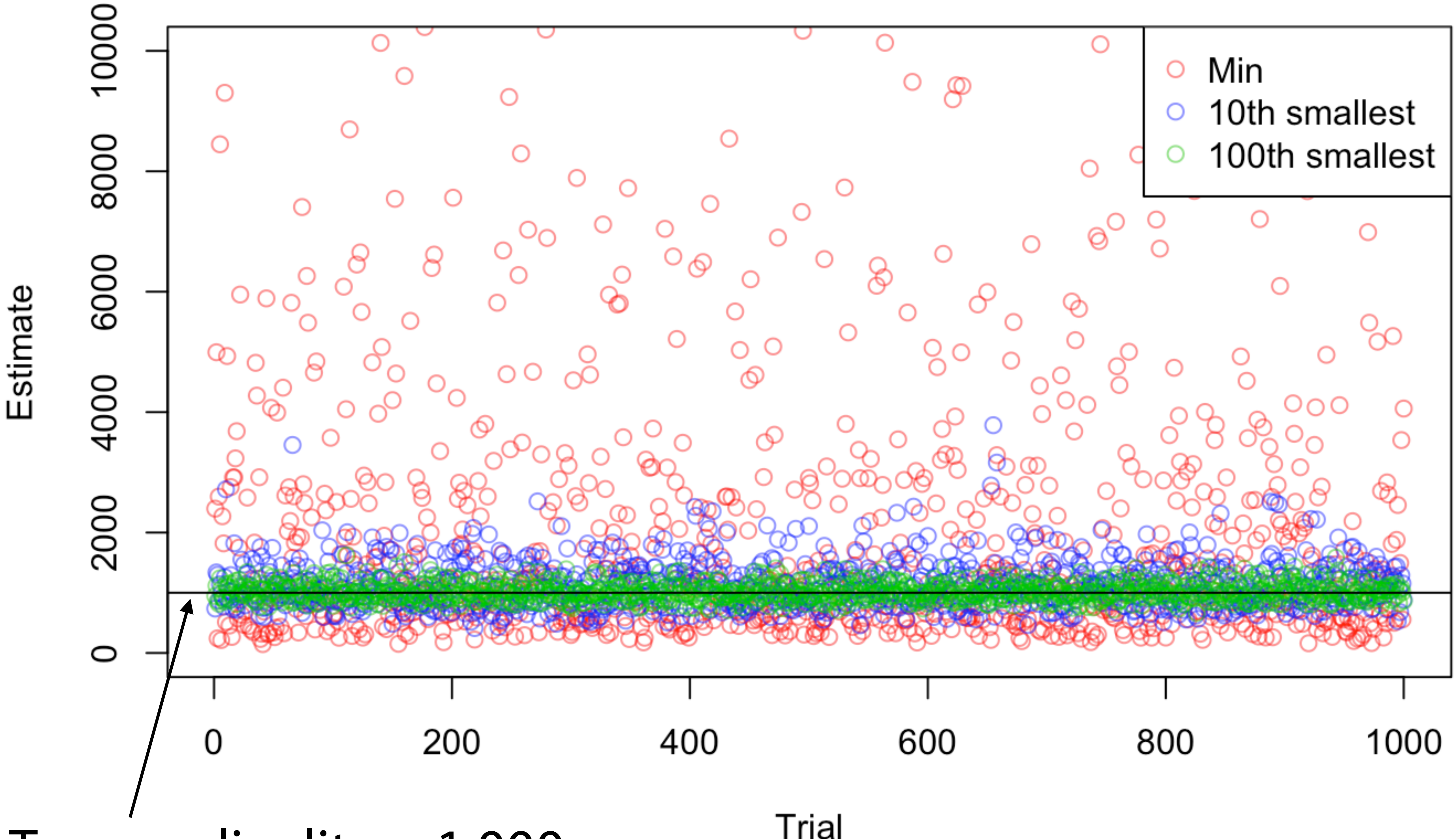
$$= \Big[\underbrace{\mathbf{E}[M_1]} + \underbrace{(\mathbf{E}[M_2] - \mathbf{E}[M_1])} + \ldots + \underbrace{(\mathbf{E}[M_k] - \mathbf{E}[M_{k-1}])}\Big] \cdot \frac{1}{k}$$



k^{th} minimum value (KMV)

Averages k estimates for $\dfrac{1}{N+1}$

Cardinality

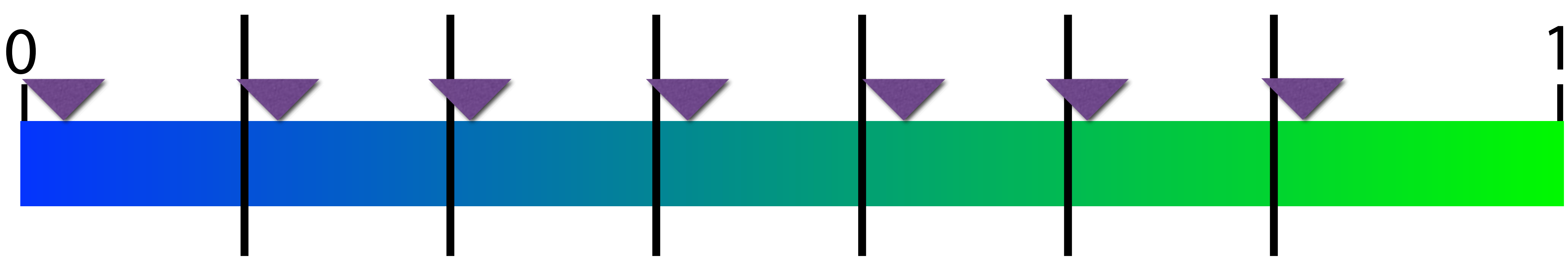


True cardinality = 1,000

Cardinality

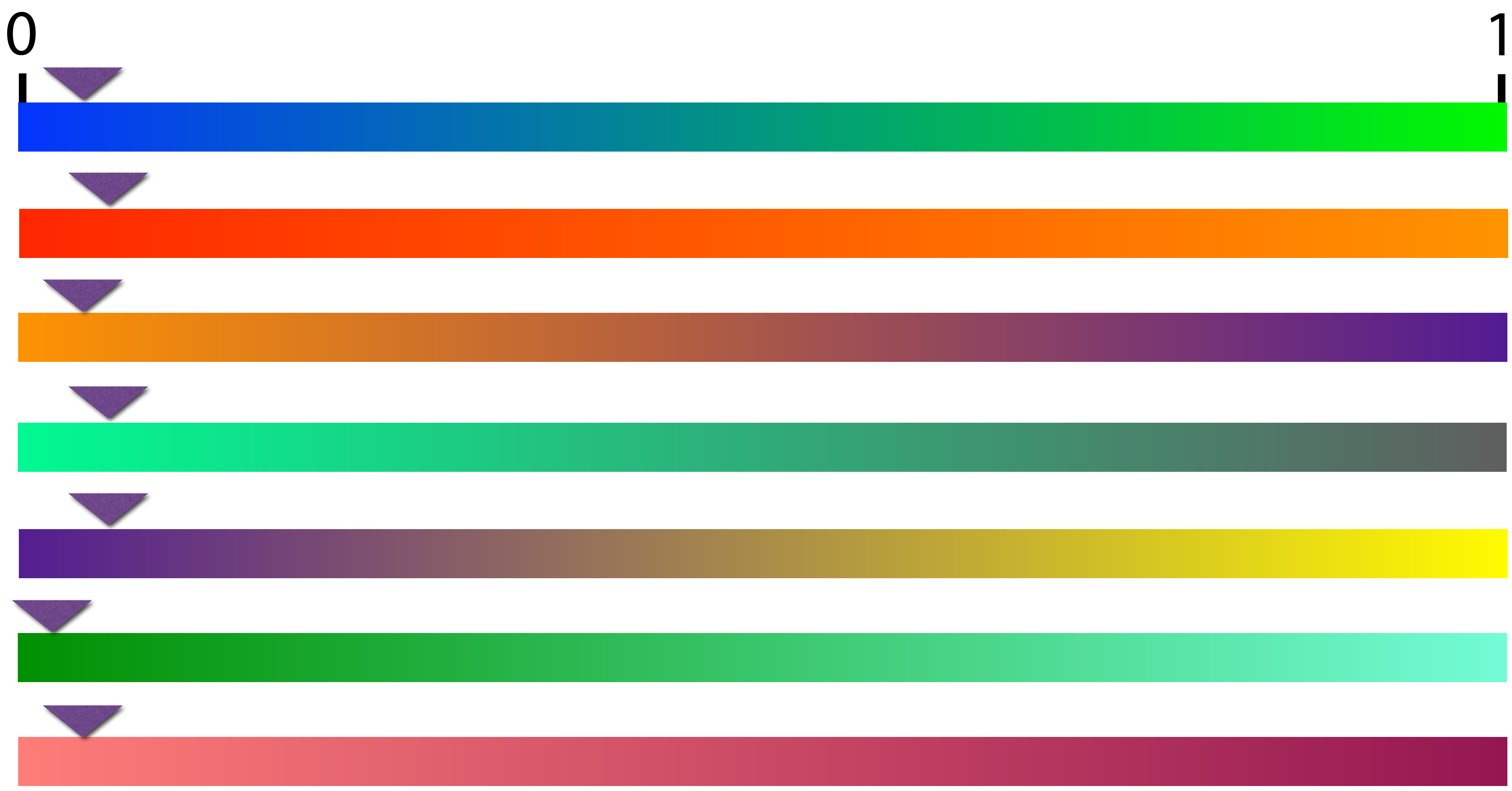
Tracking up to k minima squeezes more estimating power from a single hash function

Alternatively, we can partition the range of the function k ways and find a minimum in each



Cardinality

Or: use k hash functions, each giving a new ordering. Find minimum hash value using each.



Cardinality

k hashes

Bottom k

k partitions

All benefit from averaging. Bottom-k and k-partitions need just 1 hash function, sacrificing a little accuracy & resolution.

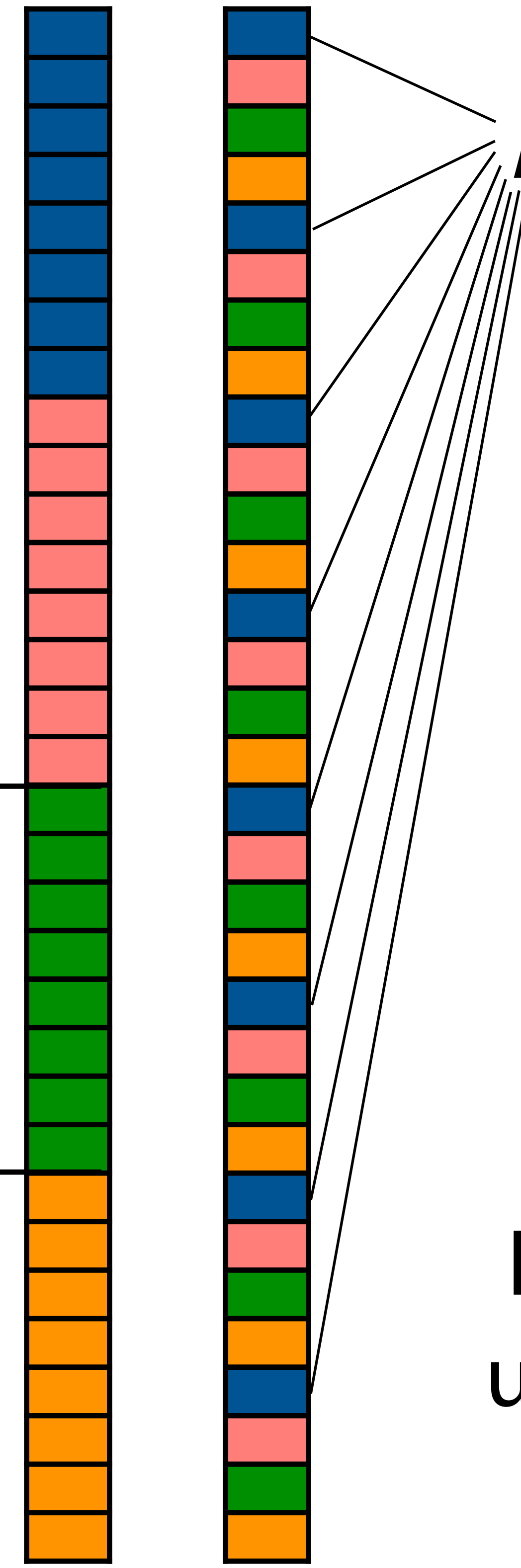
Cardinality

With a full hash table, we can simply store the set.

What if we only stored items in a ***partition*** of the table?

$$\frac{n}{2} \leq h(x) < \frac{3n}{4}$$

Where n is the range of the hash function



$$h(x) = 0 \mod 4$$

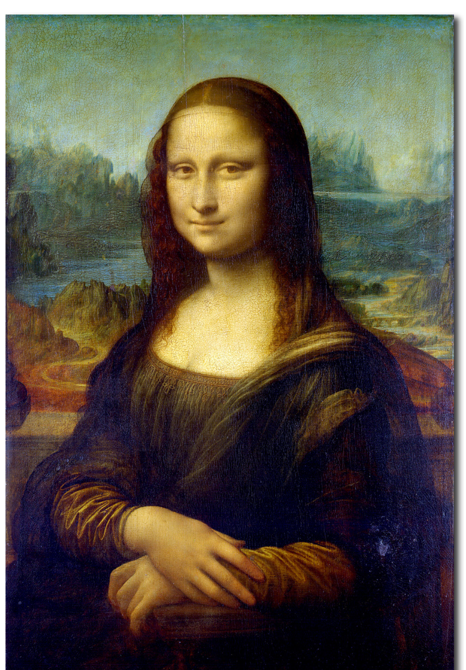
What if we kept only buckets divisible by 4?

Either way, thanks to uniform hashing, we we've uniformly sampled $1/4$ of the items

Cardinality

k^{th}-minimum-value (KMV) is a strategy for estimating the cardinality of a set

Keeping minimal hash values is also a *sketching* strategy, enabling similarity comparisons later



to be or not
to be that is
the question
whether tis
nobler in the
mind to suffer
the slings...

$h(x)$

{ 70, 112,
332, 398 }