Data Structures and Algorithms
Counting and Cardinality Sketches

CS 225
Brad Solomon

April 27, 2022

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Department of Computer Science

Last POTD today! No labs this week!

Learning Objectives

Review and finalize fundamentals of bloom filters

Discuss strategies for *counting* the occurrences of objects

Introduce the concept of cardinality and cardinality estimation

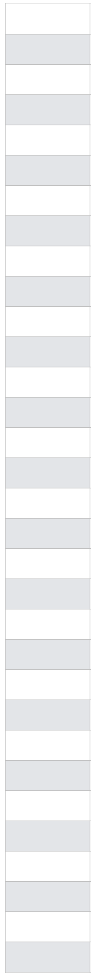
Sketch

A “sketch” is a compact (reduced) representation of a dataset that acts as a replacement for calculations.

Bloom Filters

A probabilistic data structure storing a set of values

$h_{\{1,2,3,\ldots,k\}}$



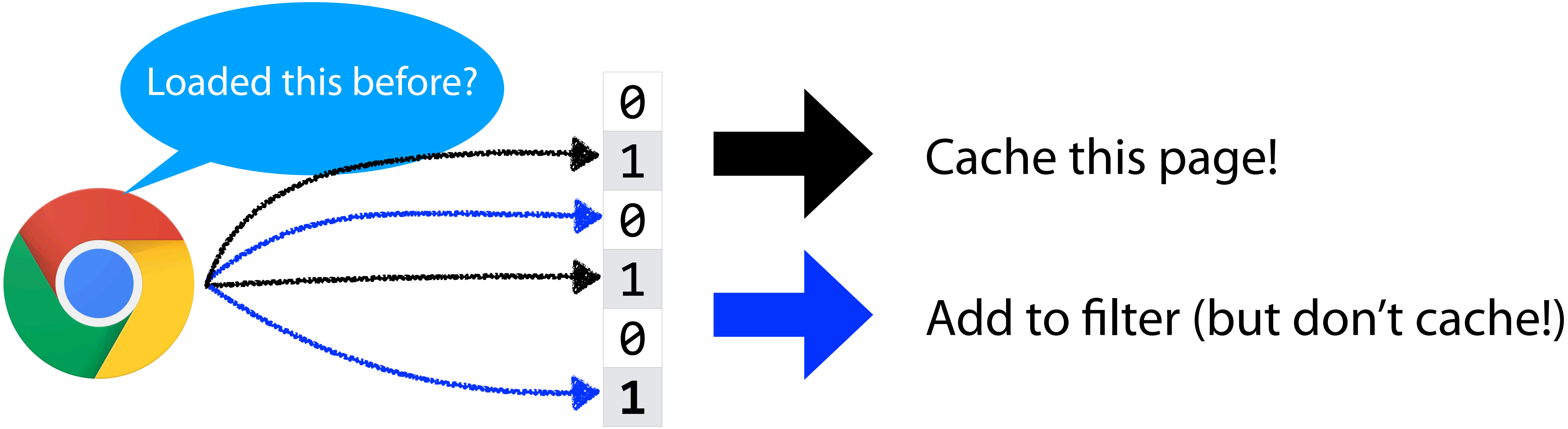
Has three key properties:

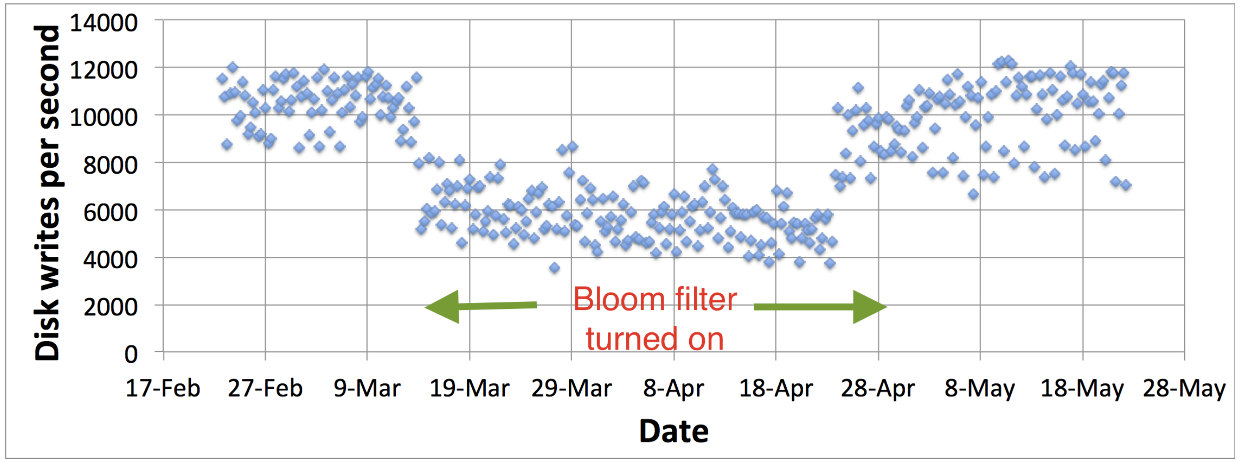
- k, number of hash functions
- n, expected number of insertions
- m, filter size in bits

Expected false positive rate: $$\boxed{\left(1-\left(1-\frac{1}{m}\right)^{nk}\right)^{k} \approx \left(1-e^{\frac{-nk}{m}}\right)^{k}}$$

Optimal accuracy when: $$\boxed{k^{*} = \ln 2 \cdot \frac{m}{n}}$$

Bloom Filter: Website Caching

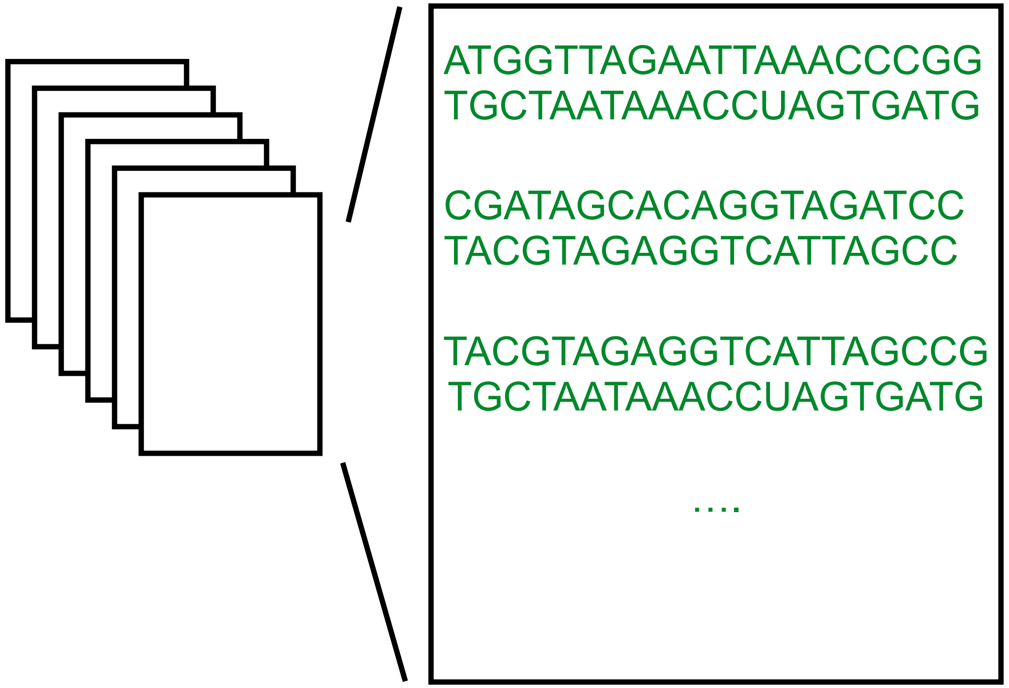




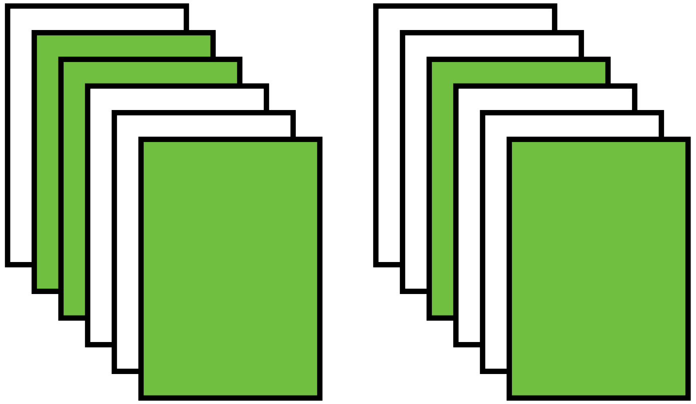
Maggs, Bruce M., and Ramesh K. Sitaraman. **Algorithmic nuggets in content delivery.** *ACM SIGCOMM Computer Communication Review* 45.3 (2015): 52-66.

Sequence Bloom Trees

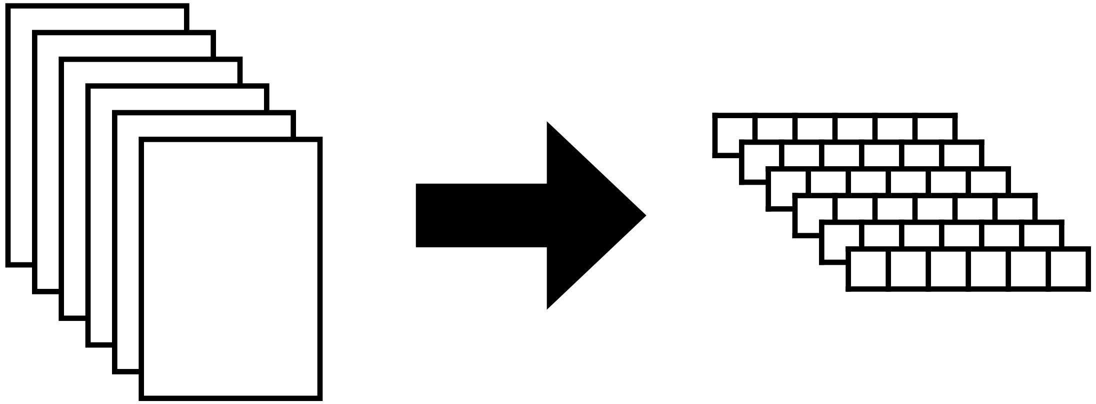
Imagine we have a large collection of text…



And our goal is to search these files for a query of interest…



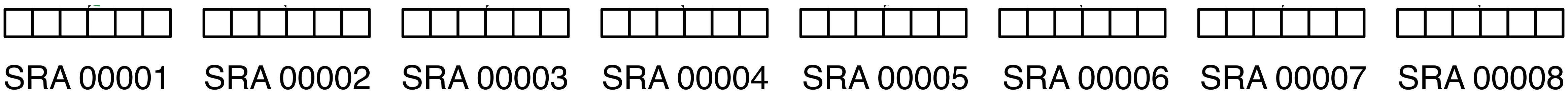
Sequence Bloom Trees



GTATGCACGCGATAG
TAGCATTGCGAGACG
TGTCTTTGATTCCTG
GACGCTGGAGCCGGA
TATCGCACCTACGTT
CACGGGAGCTCTCCA
GTATGCACGCGATAG
GCGAGACGCTGGAGC
CCTACGTTCAATATT
GACGCTGGAGCCGGA
TATCGCACCTACGTT
CACGGGAGCTCTCCA

0	1
1	0
2	1
3	1
4	0
5	0
6	1
7	0
8	0
9	1

Sequence Bloom Trees



Bloom Filters: Unioning

Bloom filters can be trivially merged using bit-wise union.

0	1
1	0
2	1
3	1
4	0
5	0
6	1
7	0
8	0
9	1

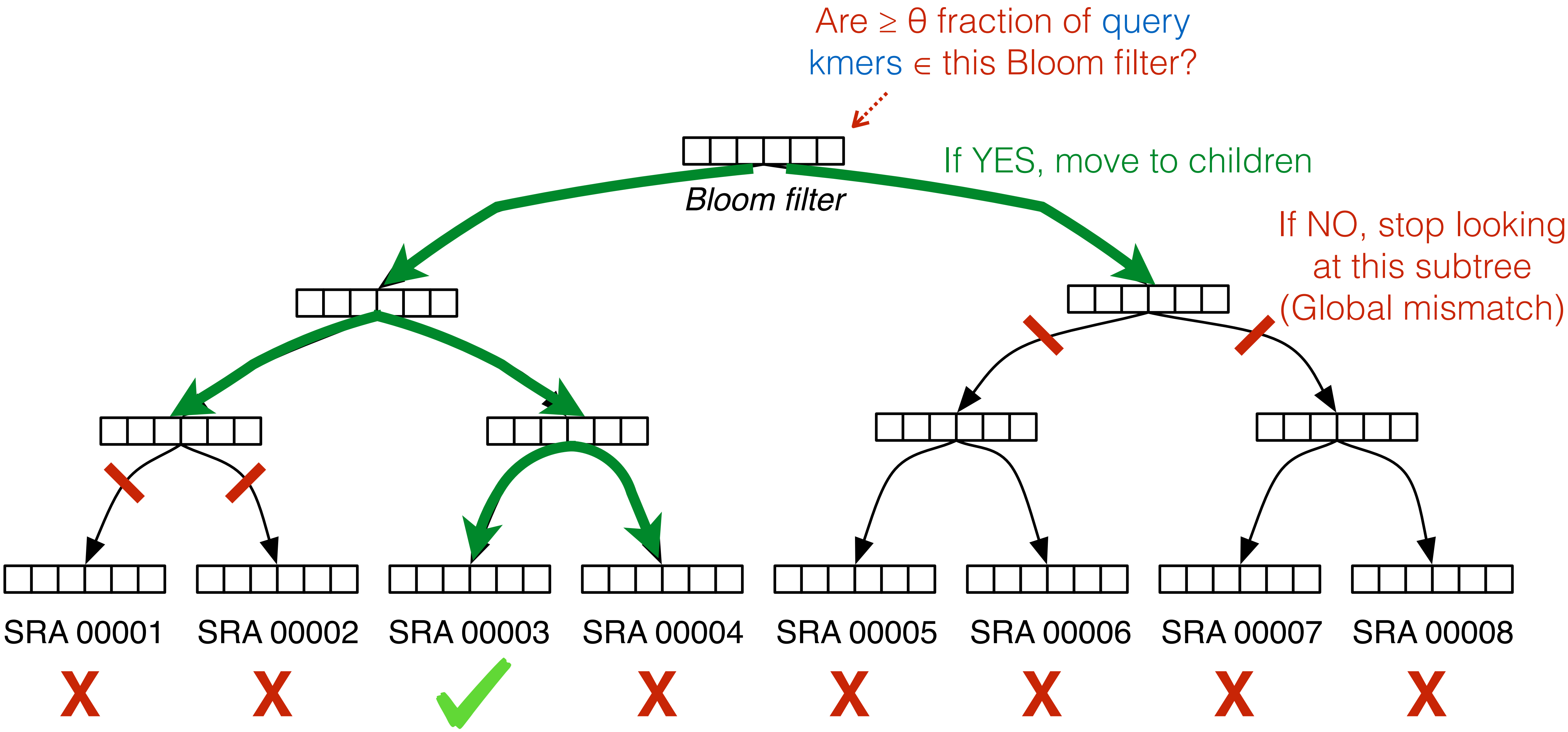
$\cup$

0	0
1	1
2	1
3	0
4	0
5	0
6	1
7	1
8	1
9	1

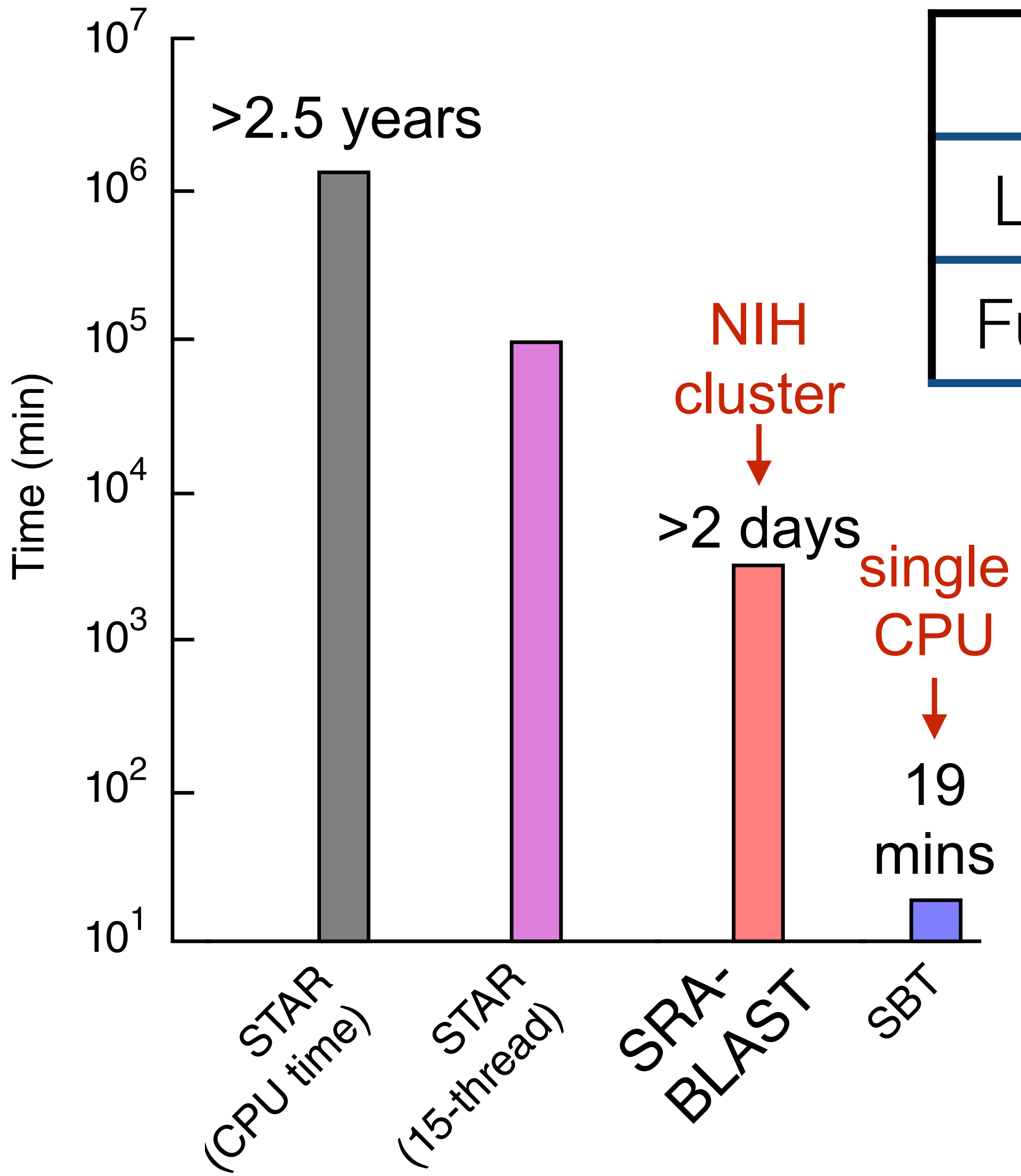
=

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Sequence Bloom Trees



Sequence Bloom Trees



	SRA	FASTA.gz	SBT
Leaves	4966 GB	2692 GB	63 GB
Full Tree	-	-	200 GB

Solomon, Brad, and Carl Kingsford. "Fast search of thousands of short-read sequencing experiments." *Nature biotechnology* 34.3 (2016): 300-302.

Solomon, Brad, and Carl Kingsford. "Improved search of large transcriptomic sequencing databases using split sequence bloom trees." *International Conference on Research in Computational Molecular Biology*. Springer, Cham, 2017.

Sun, Chen, et al. "Allsome sequence bloom trees." *International Conference on Research in Computational Molecular Biology*. Springer, Cham, 2017.

Harris, Robert S., and Paul Medvedev. "Improved representation of sequence bloom trees." *Bioinformatics* 36.3 (2020): 721-727.

Bloom Filters: Tip of the Iceberg

Cohen, Saar, and Yossi Matias. "Spectral bloom filters." *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. 2003.

Nayak, Sabuzima, and Ripon Patgiri. "countBF: A General-purpose High Accuracy and Space Efficient Counting Bloom Filter." *2021 17th International Conference on Network and Service Management (CNSM)*. IEEE, 2021.

Bonomi, Flavio, et al. "An improved construction for counting bloom filters." *European Symposium on algorithms*. Springer, Berlin, Heidelberg, 2006.

Rottenstreich, Ori, Yossi Kanizo, and Isaac Keslassy. "The variable-increment counting Bloom filter." *IEEE/ACM Transactions on Networking* 22.4 (2013): 1092-1105.

There are many more than shown here…

Counting Sketches

Sometimes we need more information than 'presence/absence'…

3201
946
5581
8945
6145
8126
3887
8925
1246
8324
4549
9100
3887
8499
8970
3921
8925
4859

Counting Bloom Filters

Instead of using one bit per register, lets use multiple!

0
1
0
1
0
0
0
1
0
1

000
110
010
001
100
110
000
000
100
111

Counting Bloom Filters

S = { 16, 8, 4, 13, 29, 11, 22 }

$h_1(k) = k \% 7$ **$h_2(k) = 2k+1 \% 7$**

0	
1	
2	
3	
4	
5	
6	

Counting Bloom Filters

S = { 16, 8, 4, 13, 29, 11, 22 }

$h_1(k) = k \% 7$ $h_2(k) = 2k+1 \% 7$

0	0
1	3
2	3
3	3
4	2
5	1
6	2

`_find(3)`

`_find(5)`

Counting Bloom Filters

S = { 16, 8, 4, 13, 29, 11, 22 }

$h_1(k) = k \% 7$ **$h_2(k) = 2k+1 \% 7$**

0	0
1	3
2	3
3	3
4	2
5	1
6	2

`_delete(8)`

`_delete(5)`

Counting Bloom Filters

A probabilistic data structure storing a set of values

Has **four** key properties:

- k, number of hash functions
- n, expected number of insertions
- m, filter size in ***registers***
- **b, number of bits per register**

Can handle deletions at the cost of allowing false negatives!

$h_{\{1,2,3,\ldots,k\}}$

000
110
010
001
100
110
000
000
100
111

Counting Bloom Filters

Pro:

Con:

Counting Bloom Filters

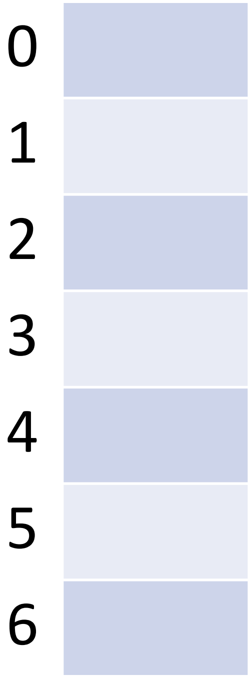
At time of insertion, what information do we have?

3
4
2
1
4
6
8
7
2
0

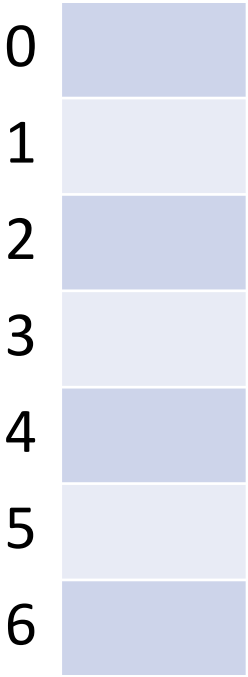
Minimal Increase

$S = \{ 1, 3, 5, 8 \}$

$h_1(k) = k \% 5$ $\quad$ $h_2(k) = 3k+1 \% 5$ $\quad$ $h_3(k) = |k - 4| \% 5$



Naive



Minimal Increase

Minimal Increase

S = { 1, 3, 5, 8 }

$h_1(k) = k \% 5$ **$h_2(k) = 3k+1 \% 5$** **$h_3(k) = |k - 4| \% 5$**

	Naive
0	3
1	4
2	0
3	3
4	2
5	0
6	0

```
_find(3)

_find(5)

_find(8)
```

	Minimal Increase
0	2
1	2
2	0
3	2
4	2
5	0
6	0

Counting Bloom Filters

Do we know anything about our collision frequency at insertion?

3
4
2
1
4
6
8
7
2
0

Spectral Bloom Filter

A counting bloom filter with two key optimizations:

1) Minimal Increase: On insert, only increment counts that have the minimum value.

2) Recurring Minimum: Insertions that have only a single minimum value *have unusually high collision likelihood!*

For these values, create a second spectral bloom filter and store them in both.

Bloom Filters: Tip of the Iceberg II

Fan, Bin, et al. "Cuckoo filter: Practically better than bloom." *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*. 2014.

Mitzenmacher, Michael. "Compressed bloom filters." *IEEE/ACM transactions on networking* 10.5 (2002): 604-612.

Crainiceanu, Adina, and Daniel Lemire. "Bloofi: Multidimensional bloom filters." *Information Systems* 54 (2015): 311-324.

Chazelle, Bernard, et al. "The bloomier filter: an efficient data structure for static support lookup tables." *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*. 2004.

There are many more than shown here…

The hidden problem with bloom filters…

Bloom Filter: Optimal Parameters

$$\boxed{k^* = \ln 2 \cdot \frac{m}{n}}$$

Given any two values, we can optimize the third

… but we often have to guess an approximate value for n!

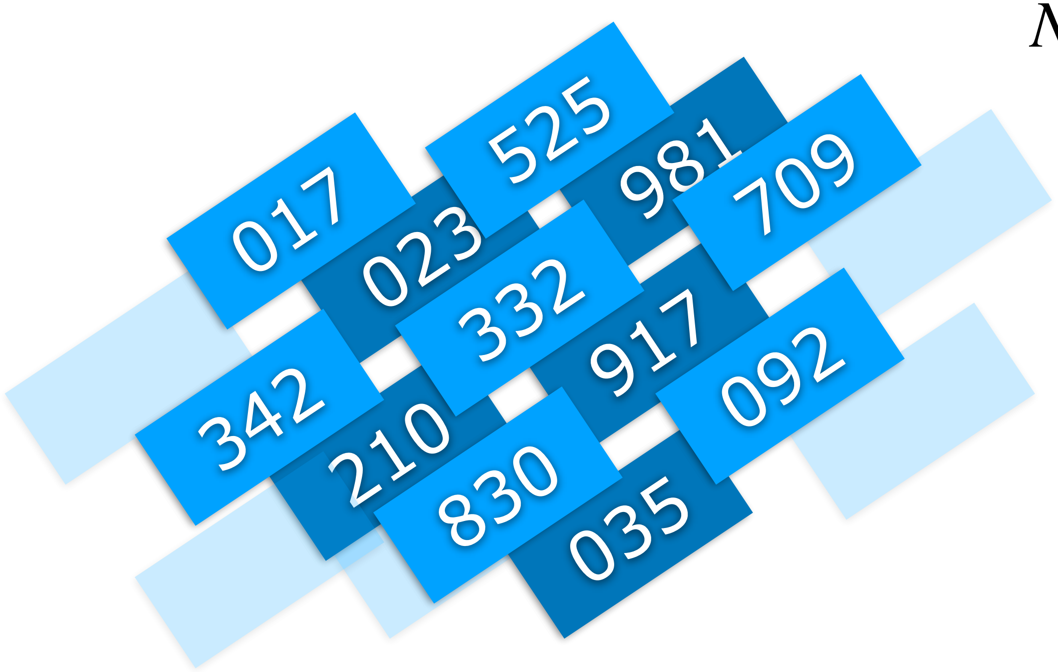
Cardinality

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
6925
2660

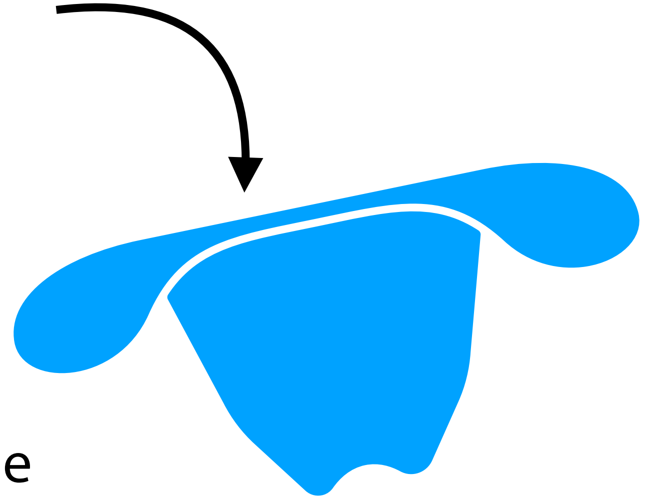
Cardinality: how many *distinct* values in a data stream?

Cardinality

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



We want to estimate N



We can see **one representative** from the cards in the hat; which to pick?

Minimum, median, maximum? Somethin else?

Cardinality

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

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

Cardinality

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

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

Conceptually: If we scatter N points randomly across the interval, we end up with $N + 1$ parts, each about $1000/(N+1)$ long

Assuming our first 'partition' is about average:

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

$$N + 1 \approx 10.5$$

$$N \approx 9.5$$

Cardinality

Now imagine we have a SUHA hash (let h_{64} be a 64-bit hash)

$$\frac{h_{64}(x)}{2^{64} - 1}$$

The randomness in the hash function turns any dataset-cardinality problem into the "hat problem"

Cardinality

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

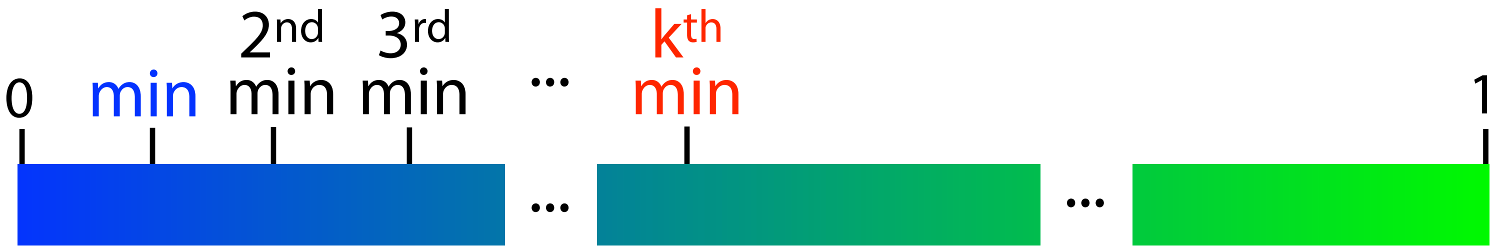
Claim: $\mathbf{E}[M] = \dfrac{1}{N+1}$

Cardinality

Attempt 1	0.455	0.220	0.951	0.236	0.979
Attempt 2	0.968	0.234	0.835	0.642	0.349
Attempt 3	0.774	0.484	0.309	0.526	0.143

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?

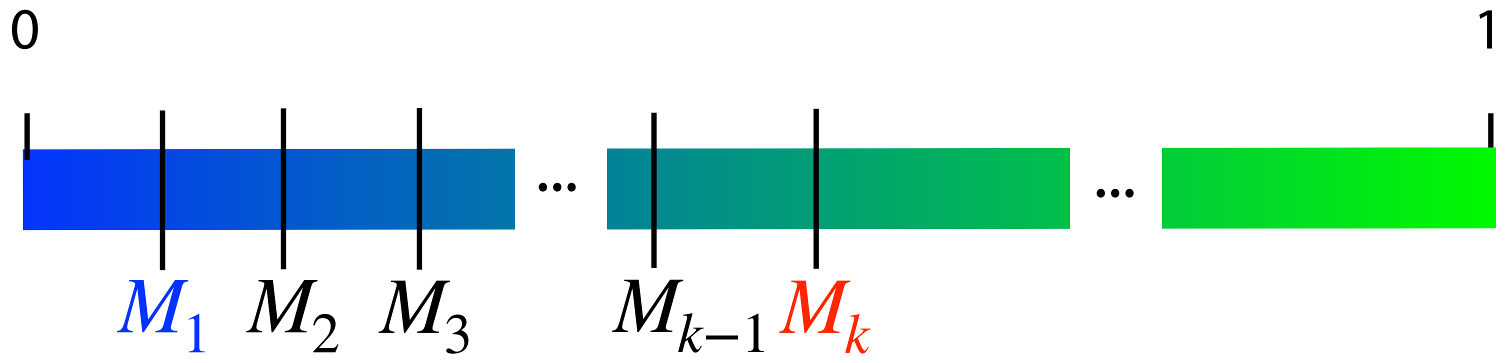
Cardinality

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

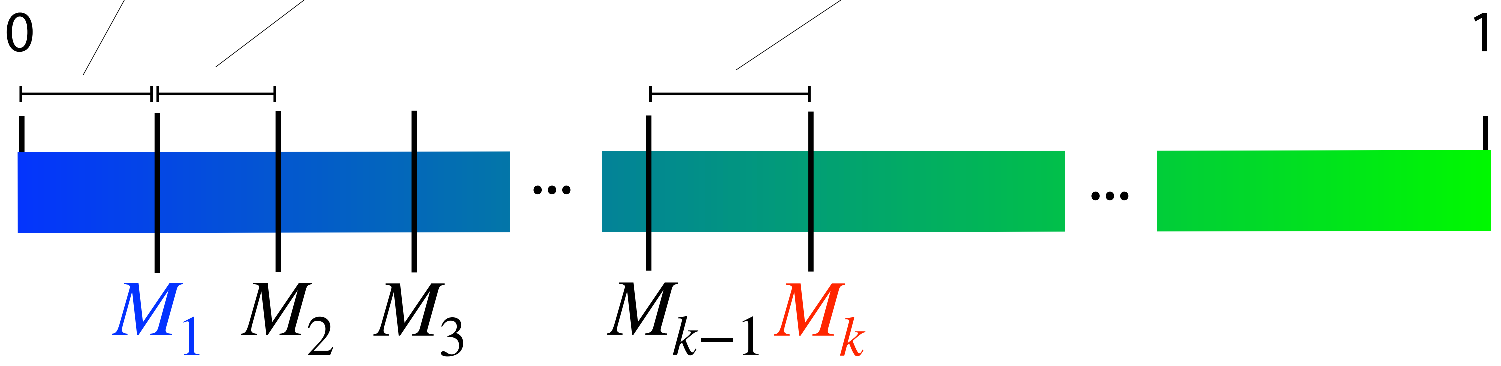
$$\frac{1}{N+1} = \frac{\mathbf{E}[M_k]}{k}$$



Cardinality

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

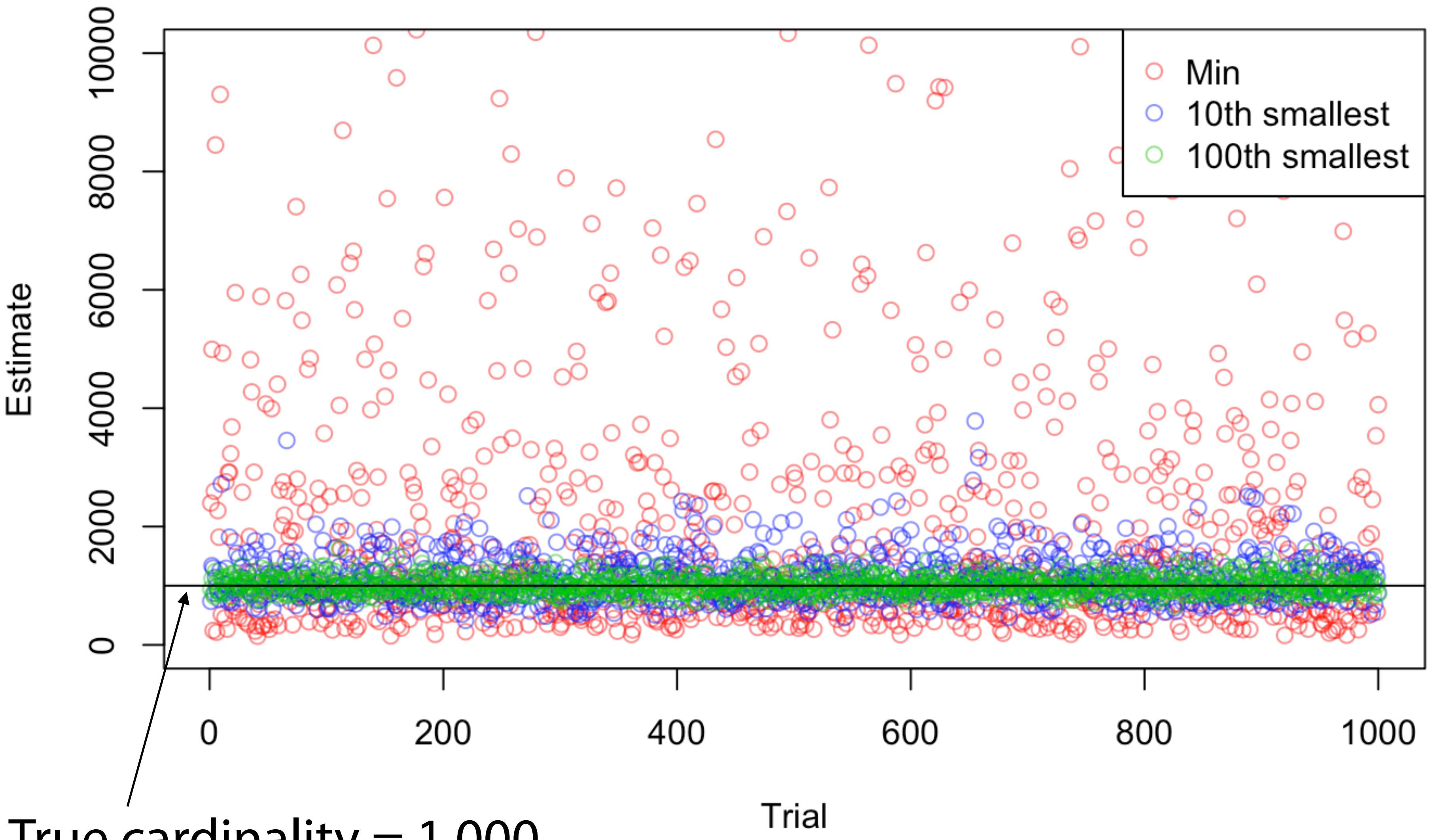
$$= \left[\mathbf{E}[M_1] + (\mathbf{E}[M_2] - \mathbf{E}[M_1]) + \ldots + (\mathbf{E}[M_k] - \mathbf{E}[M_{k-1}])\right] \cdot \frac{1}{k}$$



k^{th} minimum value (KMV)

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

Cardinality



True cardinality = 1,000