

# Data Structures and Algorithms

## Cardinality Sketches

CS 225

November 6, 2023

Brad Solomon



UNIVERSITY OF  
**ILLINOIS**  
URBANA - CHAMPAIGN

Department of Computer Science



# Learning Objectives

Finish discussing bloom filters (and review bit vectors)



Introduce the concept of cardinality and cardinality estimation

Demonstrate the relationship between cardinality and similarity

Introduce the MinHash Sketch for set similarity detection

# Bloom Filters

TP	FN
FP	TN

A probabilistic data structure storing a set of values

Has three key properties:

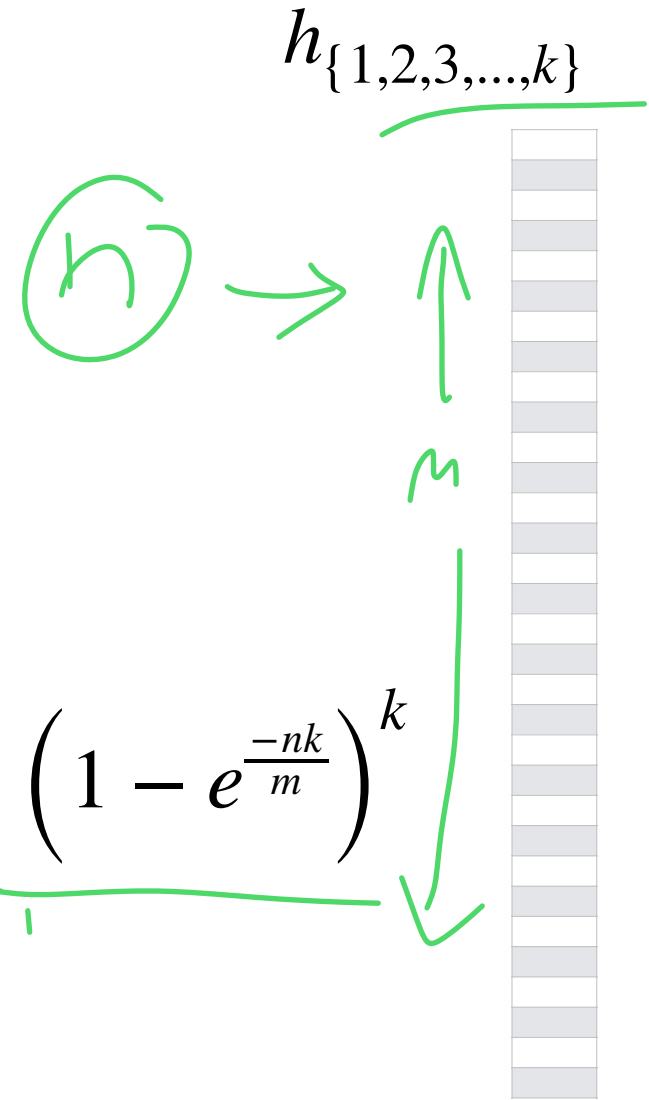
$k$ , number of hash functions

$n$ , expected number of insertions

$m$ , filter size in bits

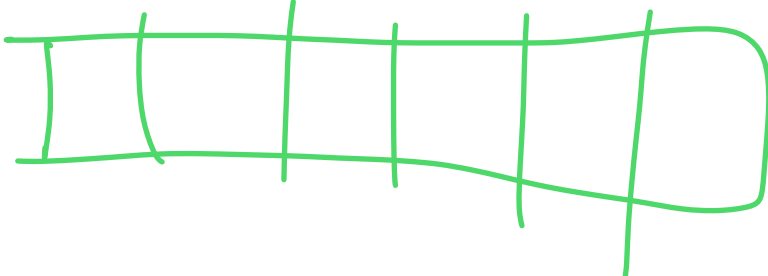
Expected false positive rate:  $\left(1 - \left(1 - \frac{1}{m}\right)^{nk}\right)^k \approx \left(1 - e^{-\frac{nk}{m}}\right)^k$

Optimal accuracy when:  $k^* = 1.44 \cdot \frac{m}{n}$



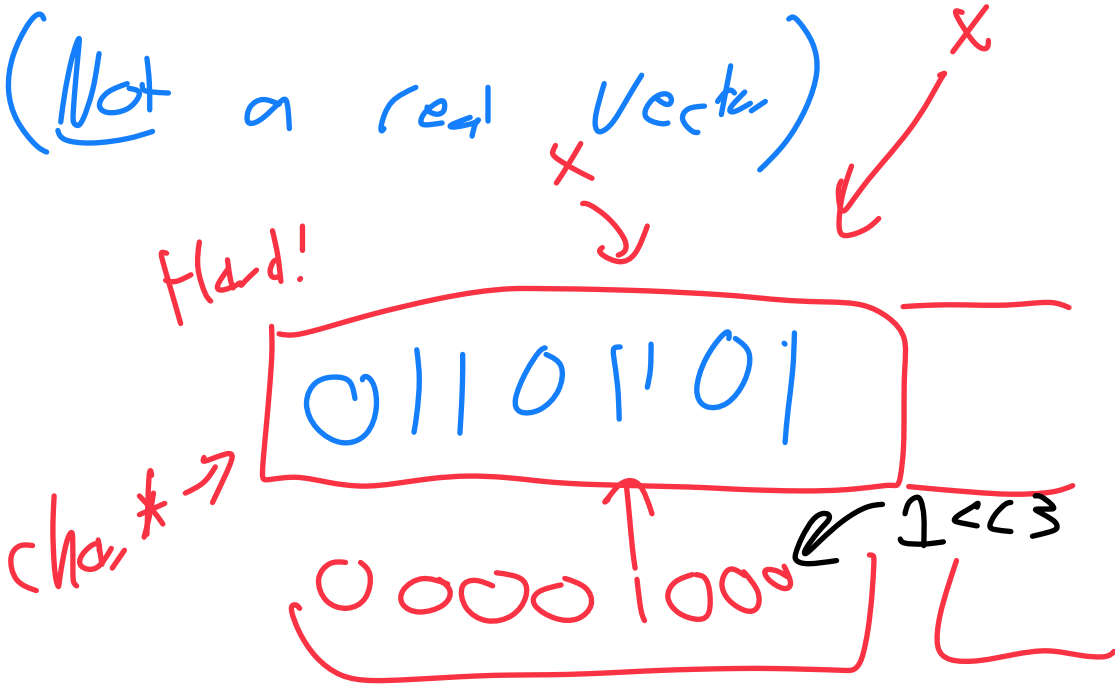
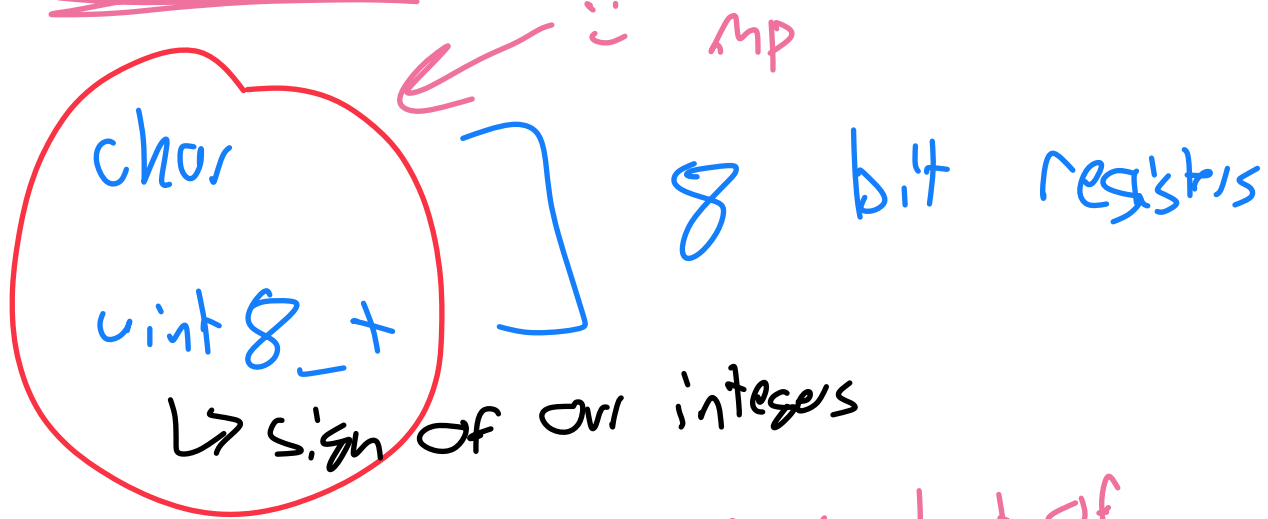
# Bitwise Operators in C++

How can we encode a bit vector in C++?



Bitset → You have to specify size at compile time

vector<bool>\* → Lab will use! (Not a real vector)



vector<int> = 0, 1 → Lot of wasted space

Bucket by 8 bits  
Bit mask

# Bitwise Operators in C++

Vector <bool> → has [] index  
char → specific b.it?

Traditionally, bit vectors are read from RIGHT to LEFT

**Warning: Lab\_Bloom won't do this but MP\_Sketching will!**

Vector <bool>      Standard indexing

Bit vectors (Binary)

A

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

←  
 $2^2$   
 $4 + 2 + 1 = 7$

Numeric

1      000...1  
2      0...10

↑  
1 << 1  
c < 2

1	0	0	1	0	1	0
---	---	---	---	---	---	---

6      5      4      3      2      1      0  
 $2^6 \rightarrow 64$        $2^3 \rightarrow 8$        $2^1 \rightarrow 2$

$4 = 100$

$64 + 8 + 2 = 74$

# Bitwise Operators in C++

Let **A = 10110**

Let **B = 01110**

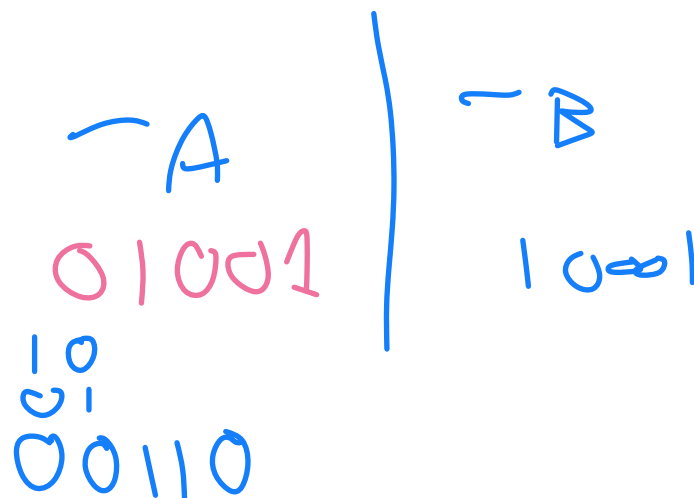
$\sim B$ : Complement (Negation)

A & B: And Operator

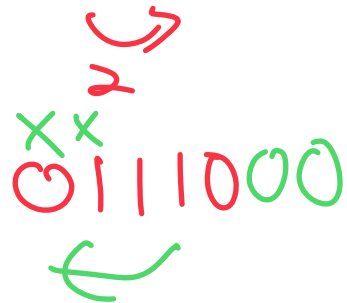
A | B: Or Operator

A >> 2: Right Shift

B << 2: Left Shift



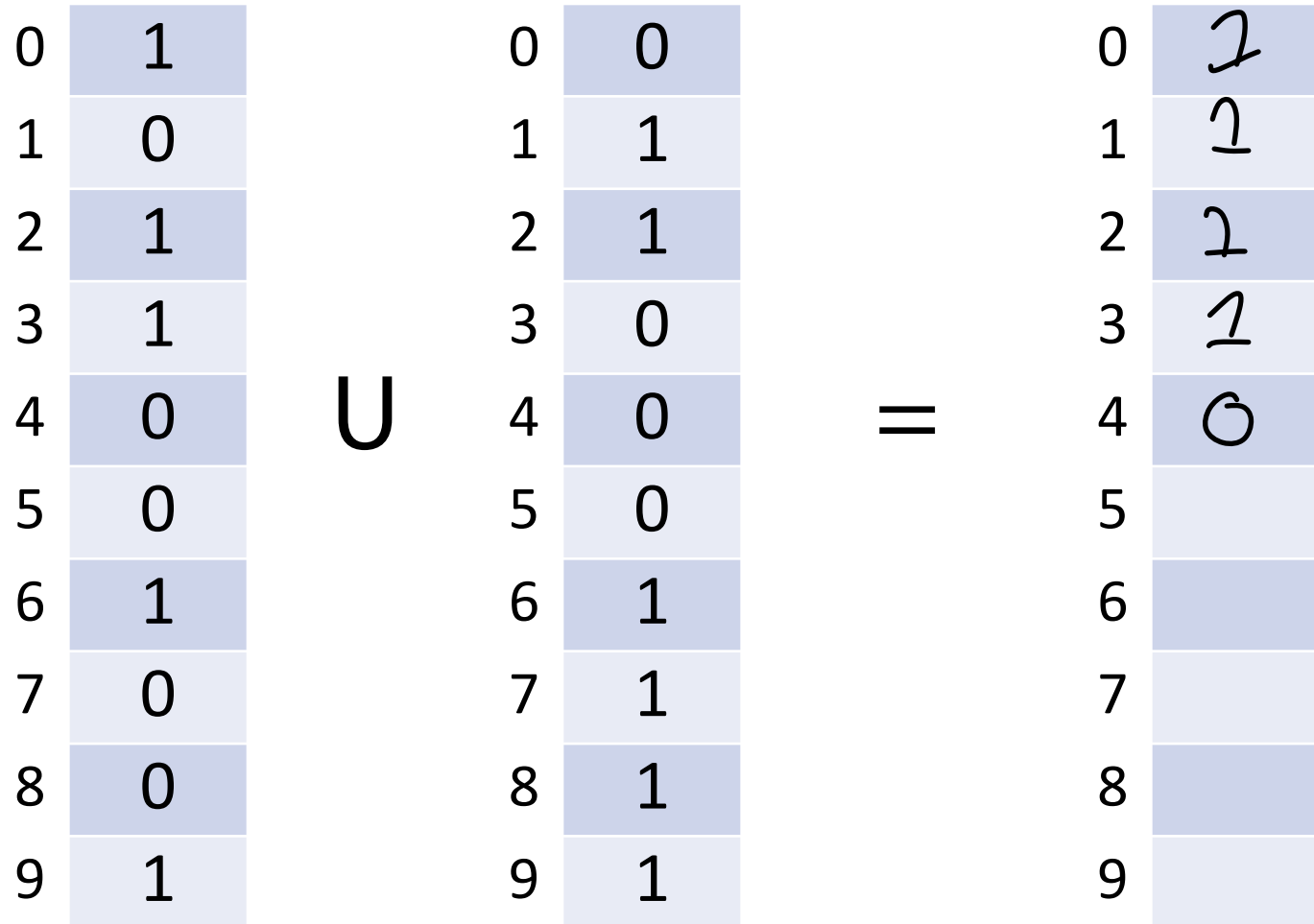
A >> 5  
400000



why would I want this?  
Division or Multiplication

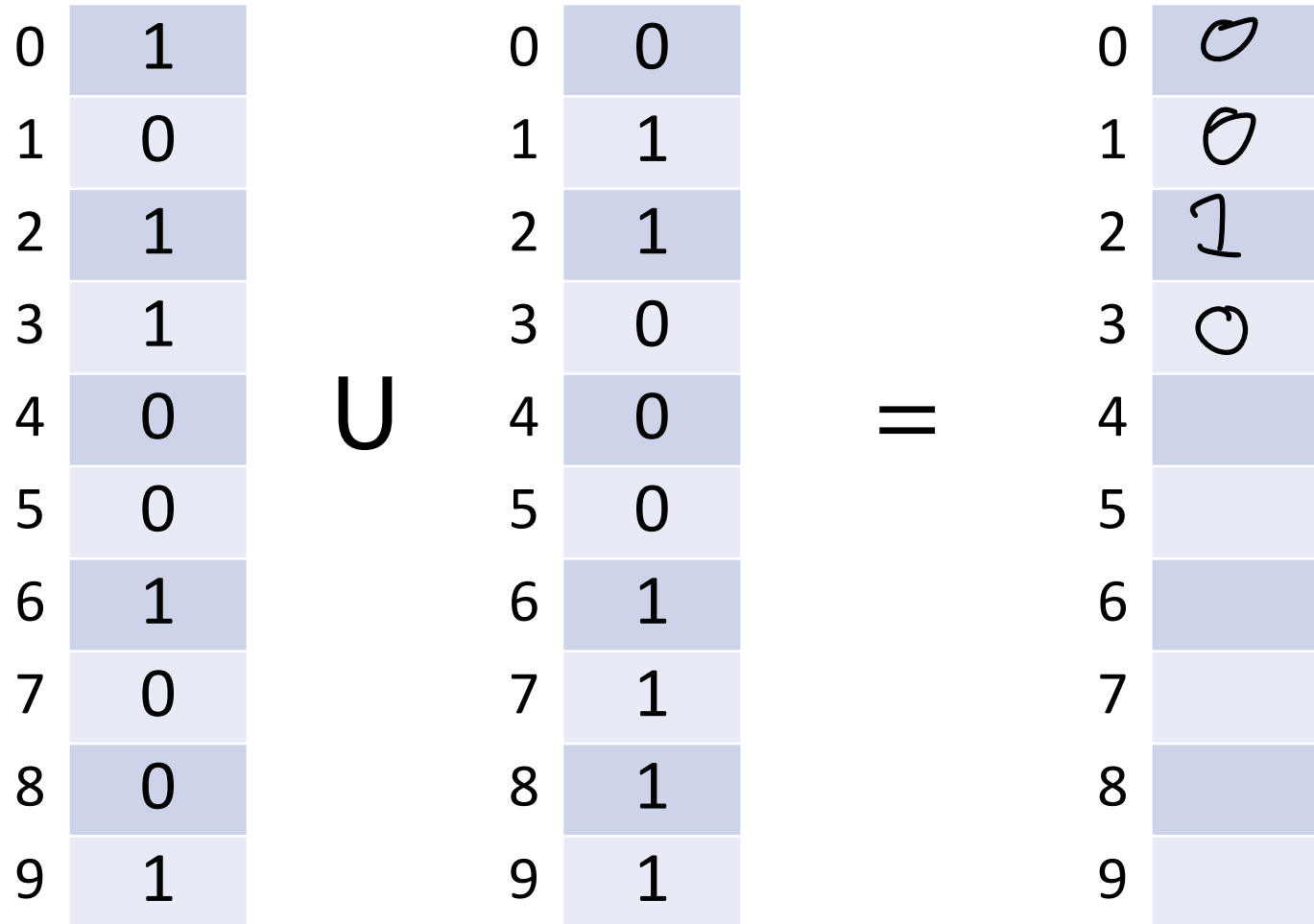
# Bit Vectors: Unioning

Bit Vectors can be trivially merged using bit-wise union.



# Bit Vectors: Intersection

Bit Vectors can be trivially merged using bit-wise intersection.





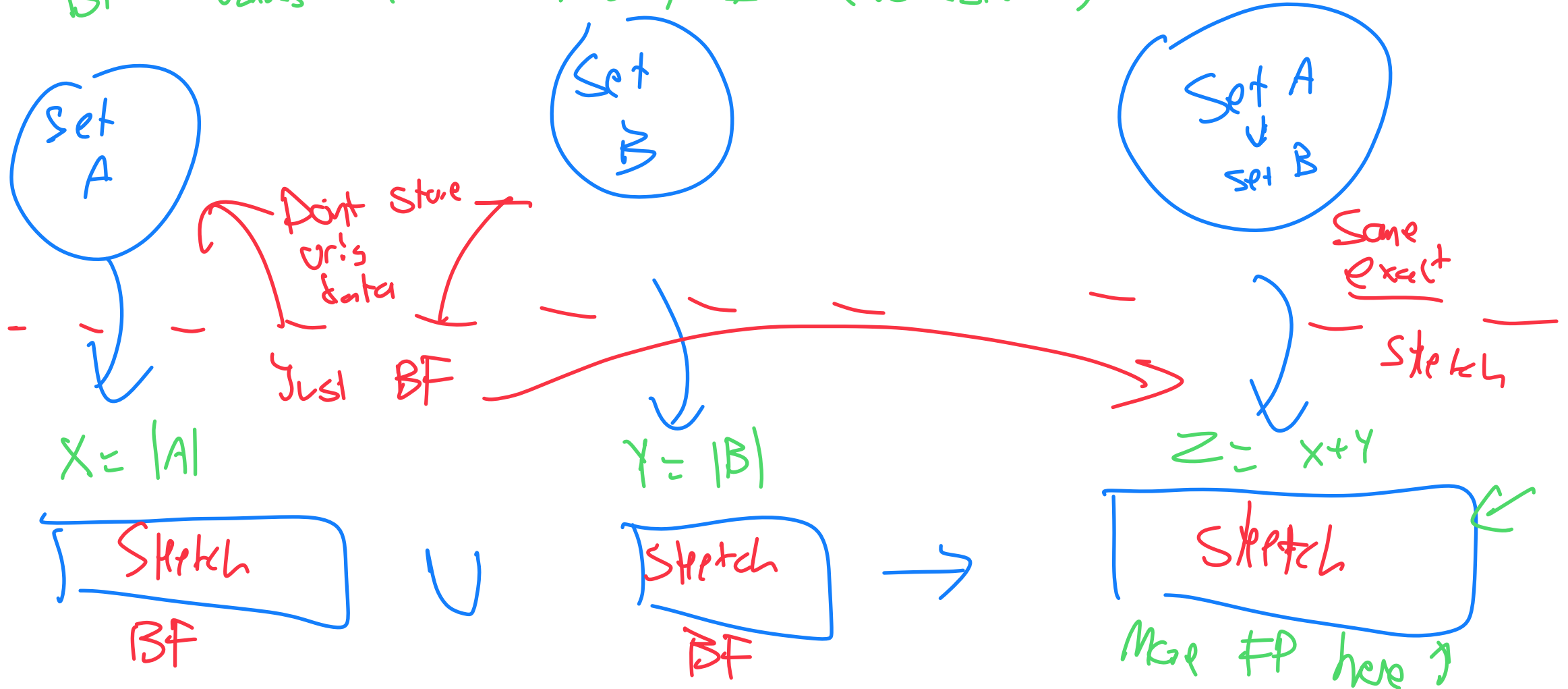
# Bit Vector Merging

Bloom Filters

(lossy transform)

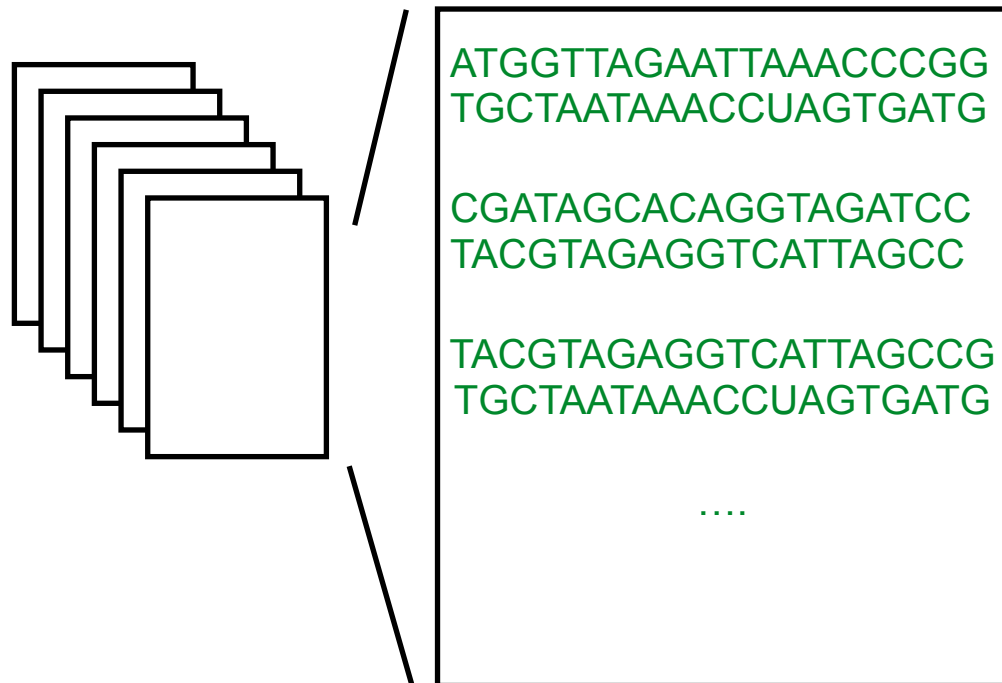
What is the conceptual meaning behind **union** and **intersection**?

BF = values that are 1 stay 1 (No collisions)

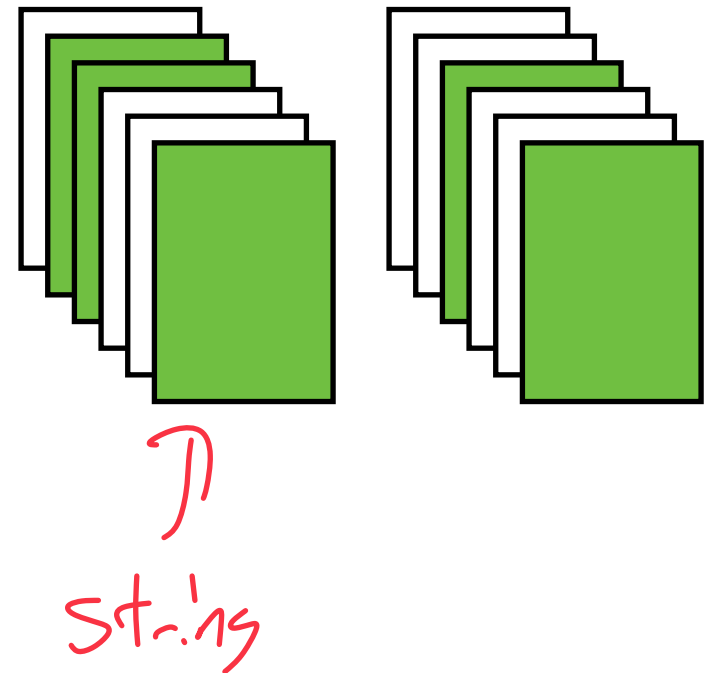


# Sequence Bloom Trees

Imagine we have a large collection of text...



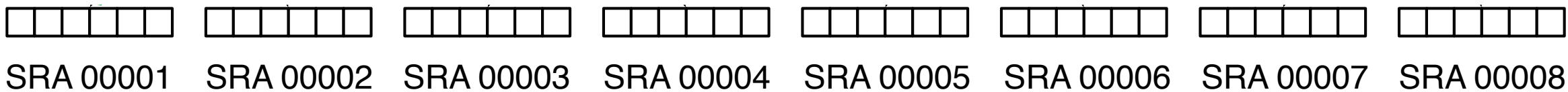
And our goal is to search these files for a query of interest...



# Sequence Bloom Trees

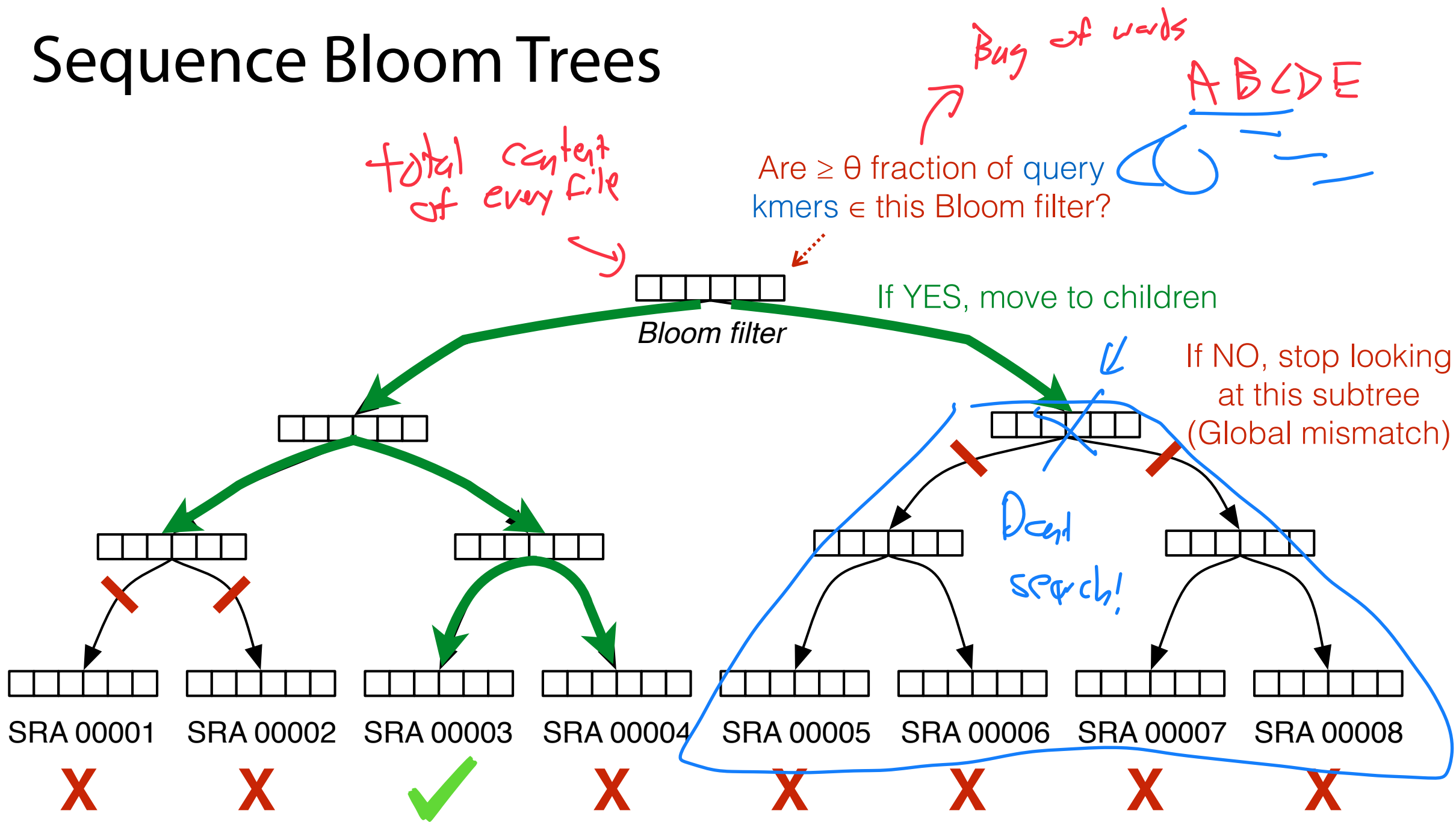
Bitwise union is fast

fast!

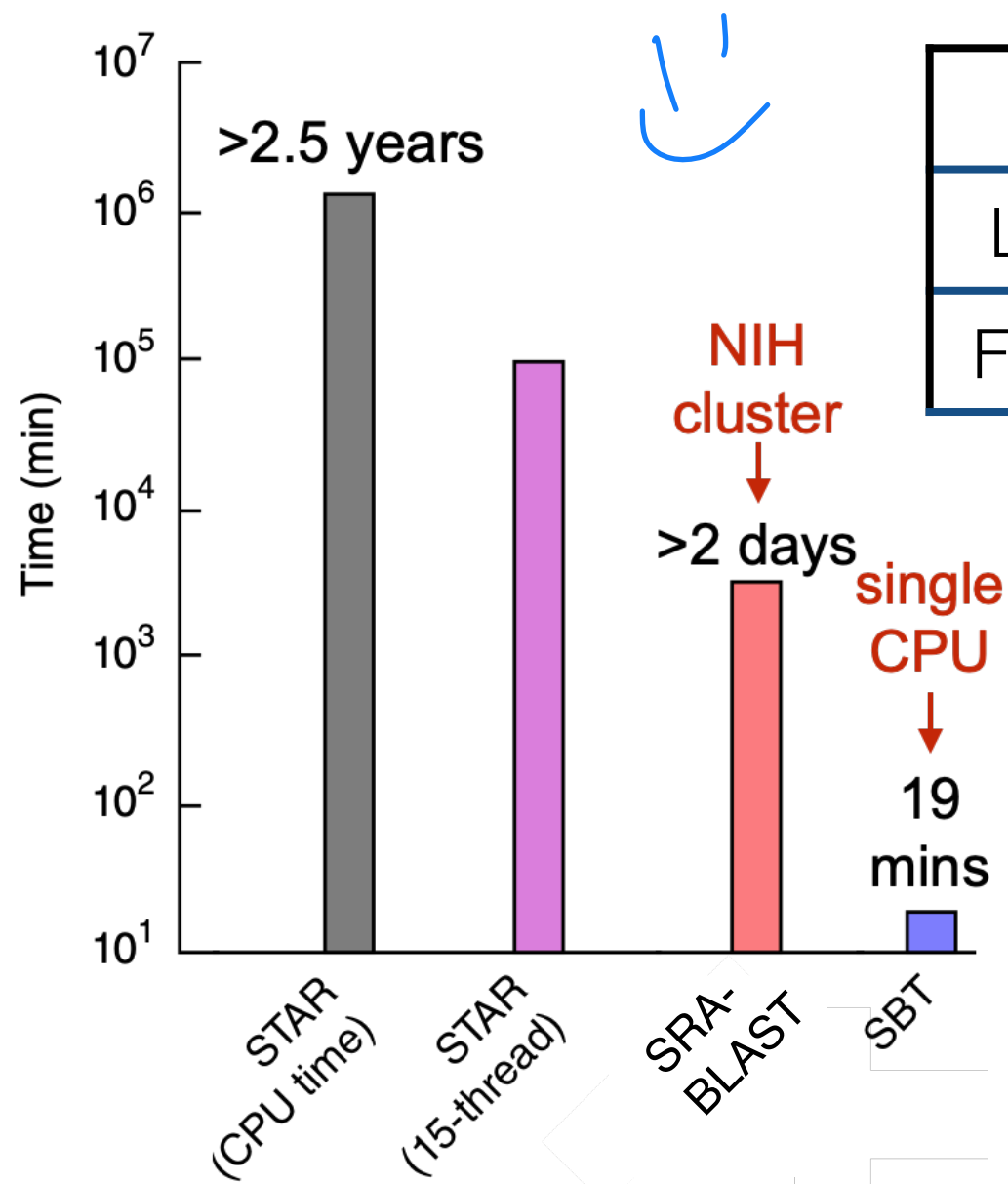


BF for every file  $O(n)$

# 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.

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.

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.

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.

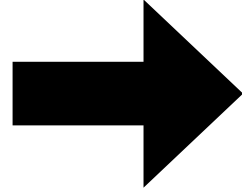
Chazelle, Bernard, et al. "The bloomier filter: an efficient data structure for static support lookup tables." *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*. 2004.

There are many more than shown here...

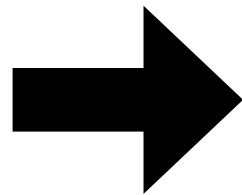
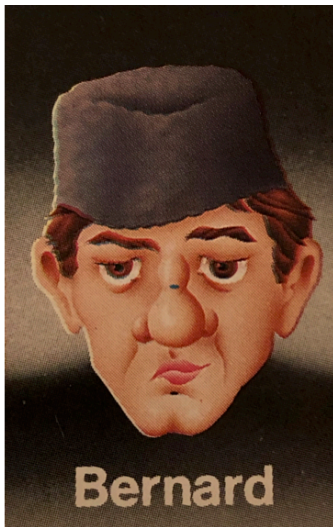
# The hidden problem with (most) sketches...

↳ BF constructed w/ fixed  $m$

↳ Not resizable



Not



↳ Super Saturated

∴ More sequences than normal!

# Cardinality

**Cardinality** is a measure of how many unique items are in a set

↳ Review interview question

↳ Hash table!

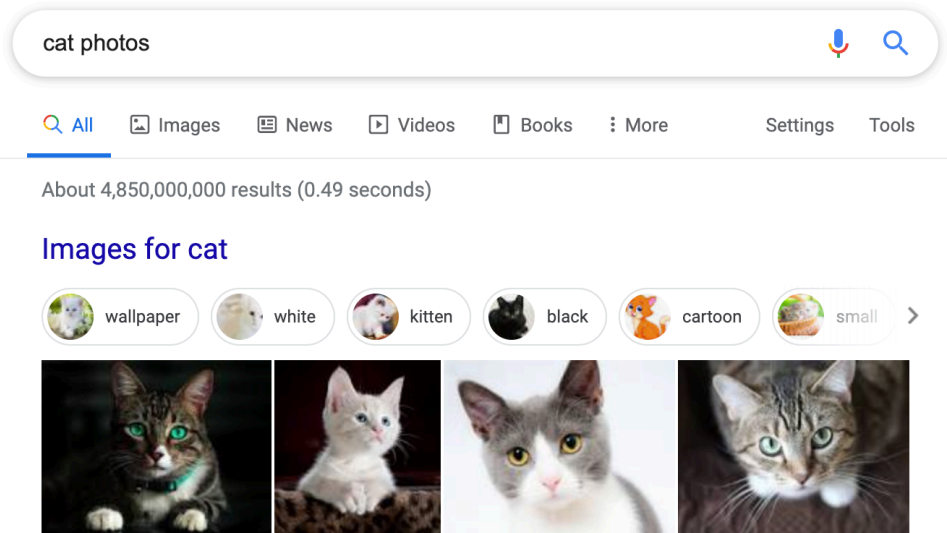
2
4
9
3
7
9
7
8
5
6



# Cardinality

Sometimes its not possible or realistic to count all objects!

Data Stream  
↳



Estimate: 60 billion — 130 trillion

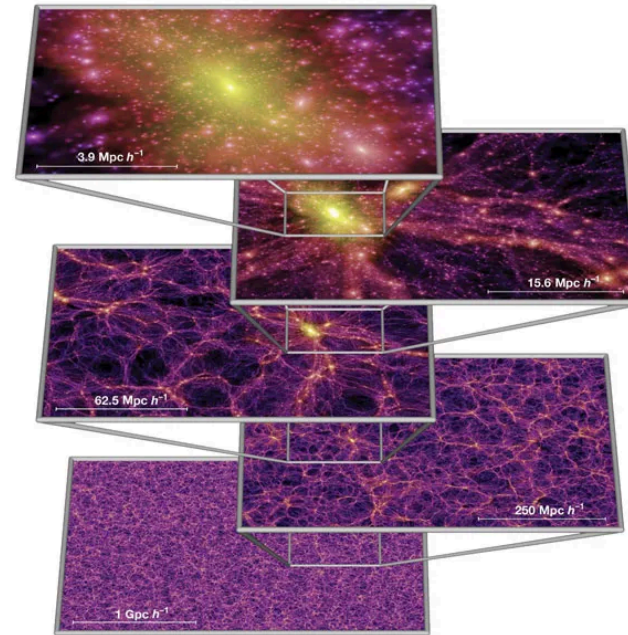


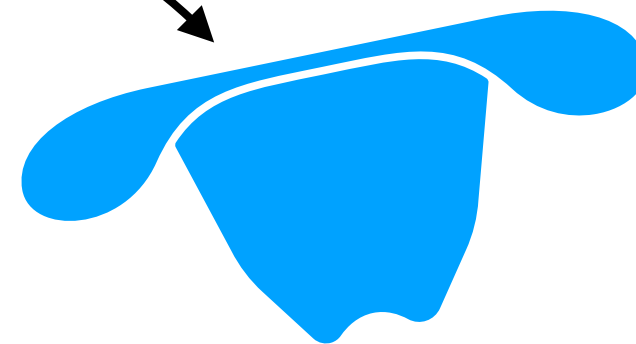
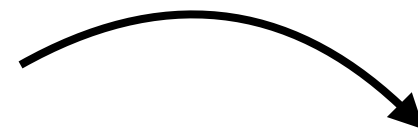
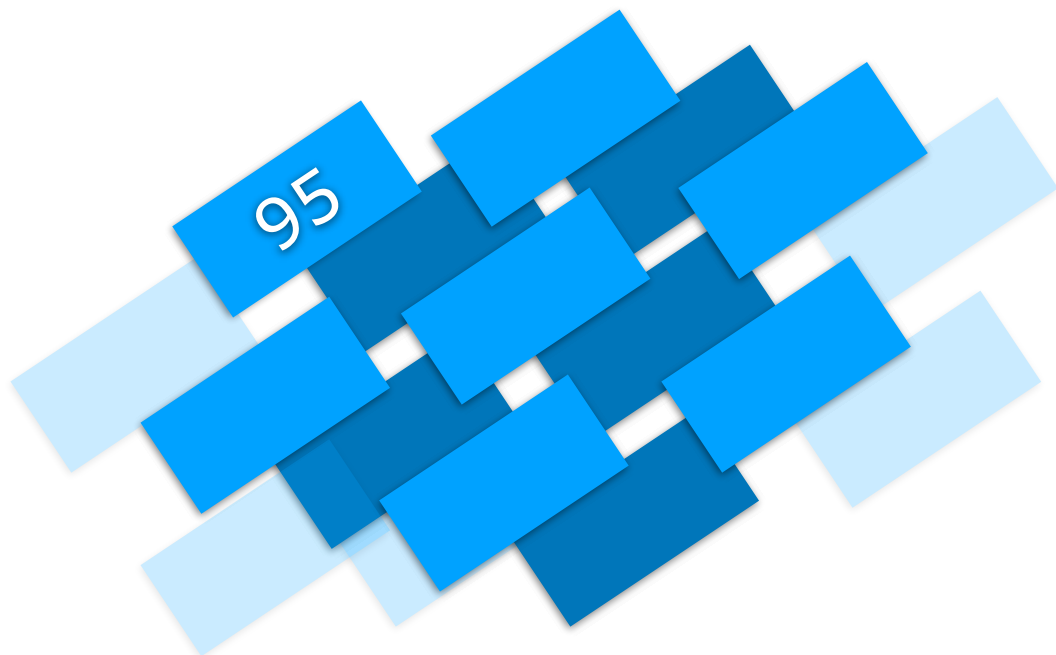
Image: <https://doi.org/10.1038/nature03597>

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
2314

# Cardinality Estimation

Imagine I fill a hat with numbered cards and draw one card out at random.

If I told you the value of the card was 95, what have we learned?



Analogy from Ben Langmead



# Cardinality Estimation

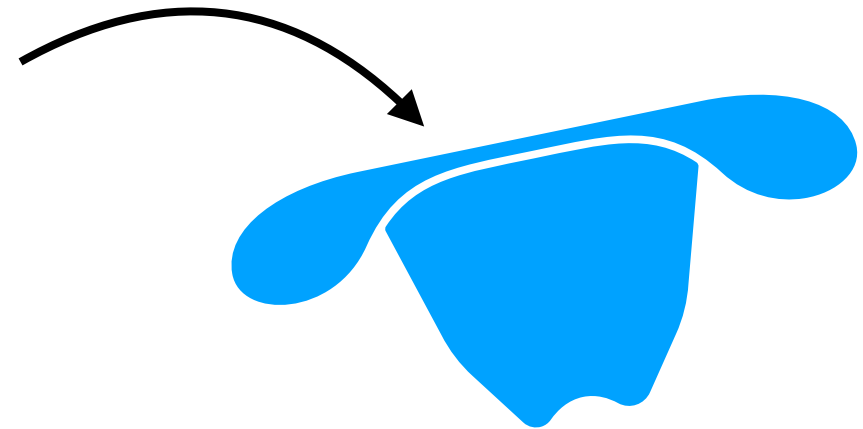
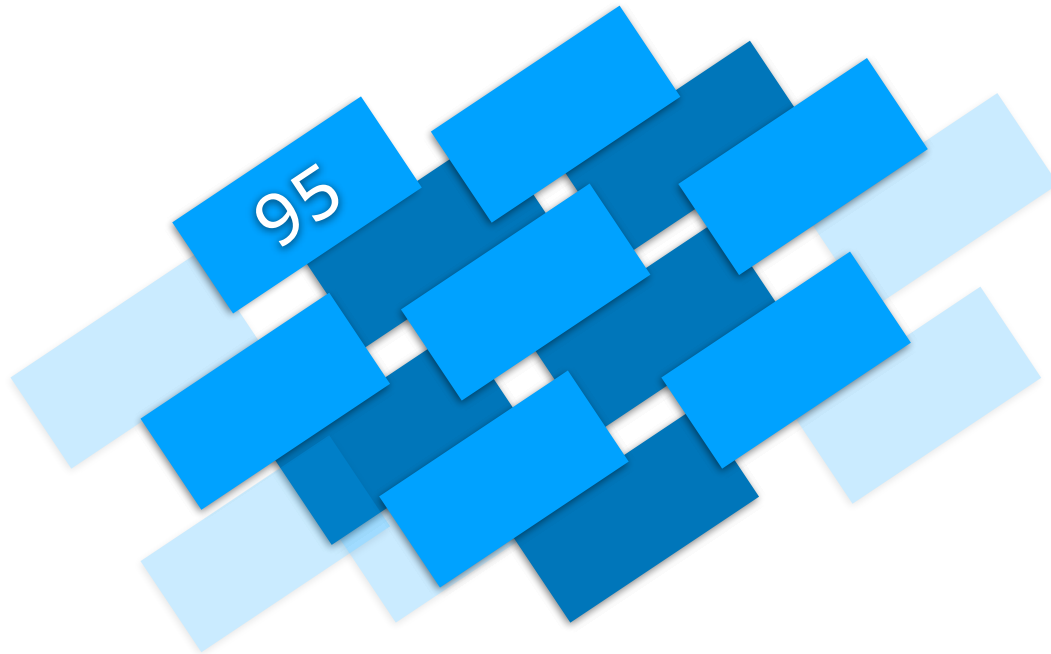
*uniform independent*

Imagine I fill a hat with a **random subset** of numbered cards **from 0 to 999**

If I told you that the **minimum** value was 95, what have we learned?

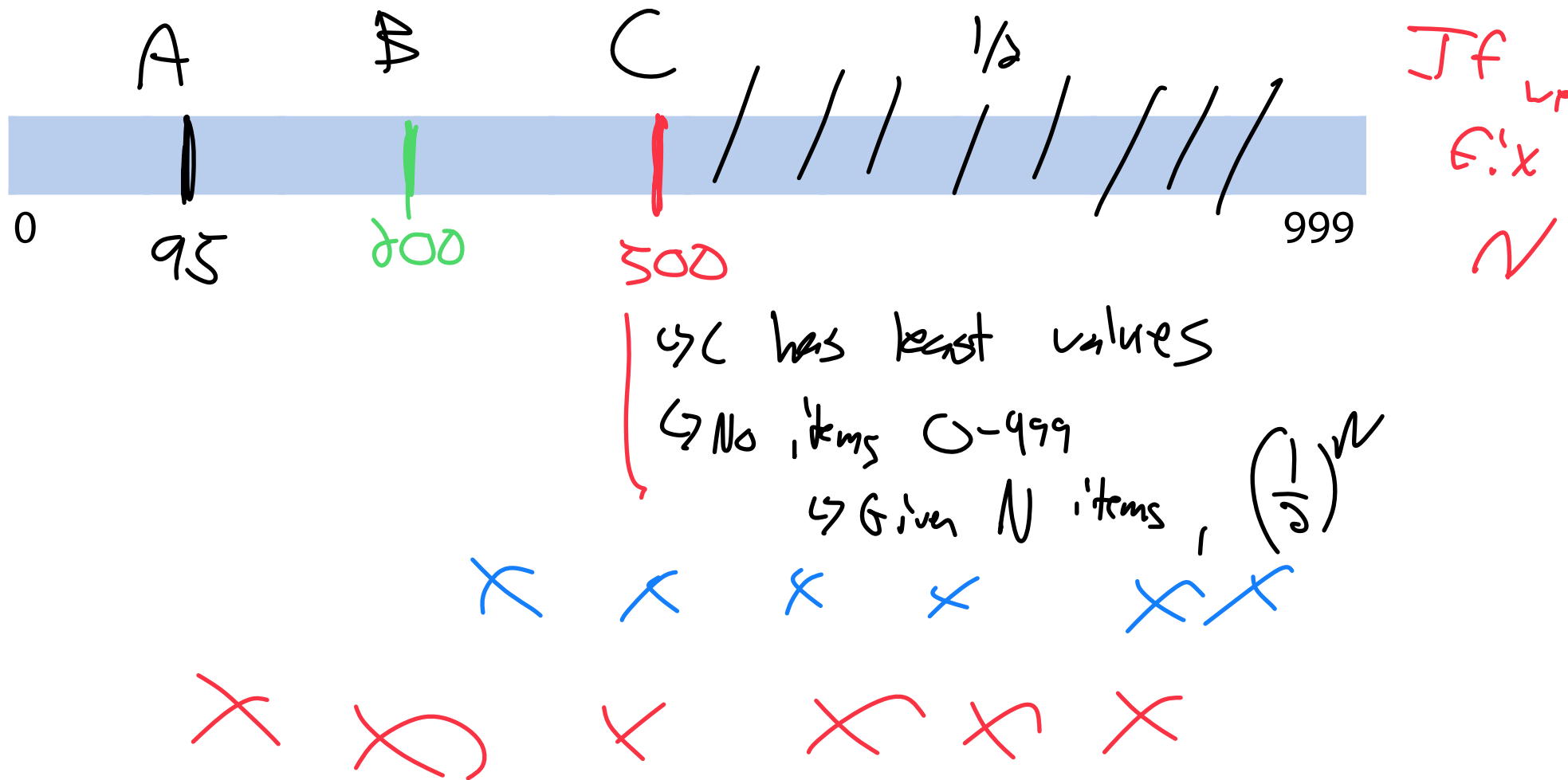
*↳ 95 tells us no items 0-94*

*↳ 10 items in set*



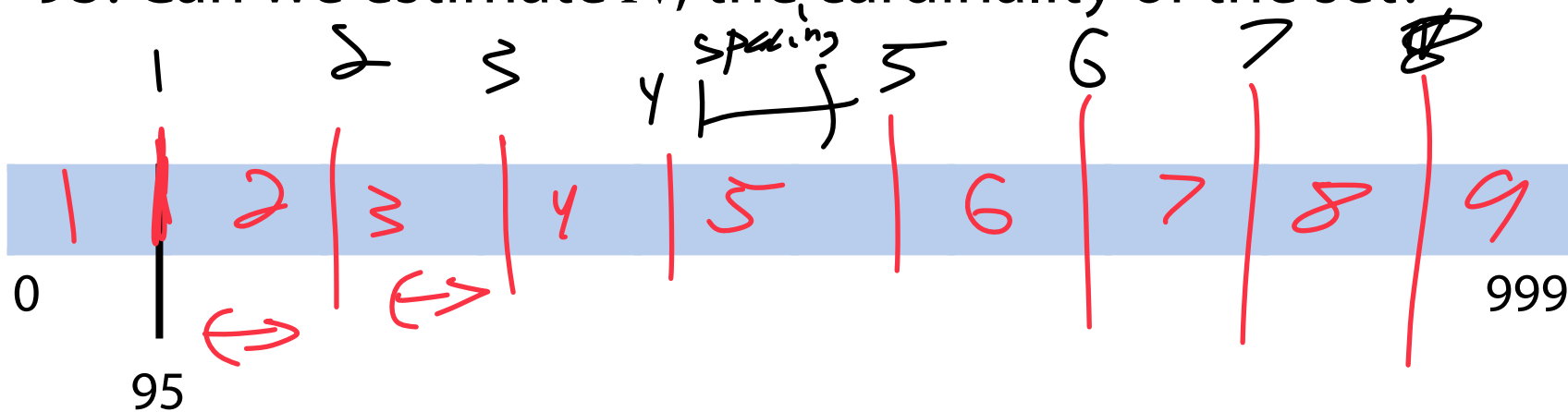
# Cardinality Estimation

Imagine we have multiple uniform random sets with different minima.



# Cardinality Estimation *uniform independent?*

Let  $\min = 95$ . Can we estimate  $N$ , the cardinality of the set?



$$95 \approx \frac{1000}{N+1} \Rightarrow N+1 \approx \frac{1000}{95}$$

$$N \approx \frac{1000}{95} - 1 \approx 9.5$$

# Cardinality Estimation



Let  $\text{min} = 95$ . Can we estimate  $N$ , the cardinality of the set?



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

Assuming our first 'partition' is about average:  $95 \approx 1000/(N + 1)$

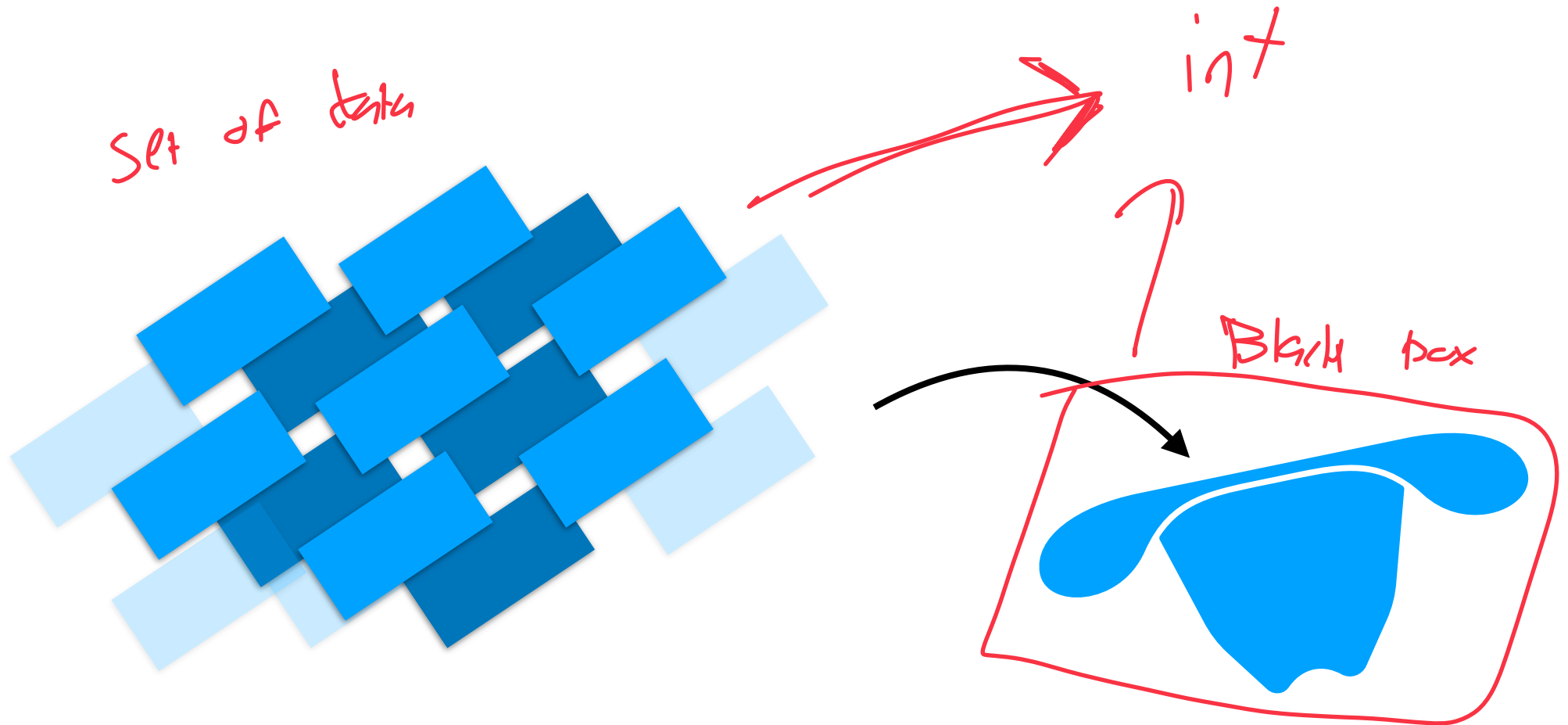
$$N + 1 \approx 10.5$$

$$N \approx 9.5$$

# Cardinality Estimation

Why do we care about "the hat problem"?

1) Est cardinality before building sketch







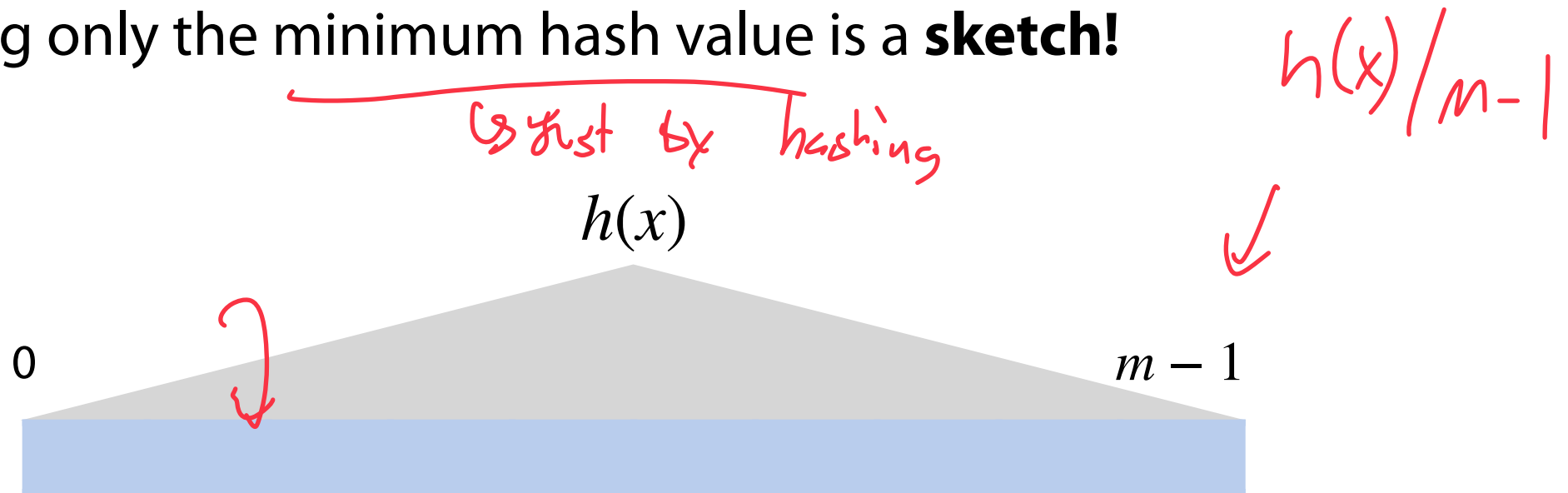


# Cardinality Estimation

Now imagine we have a SUHA hash  $h$  over a range  $m$ .

Here a hash insert is equivalent to adding a card to our hat!

Now storing only the minimum hash value is a **sketch**!



# Cardinality Sketch

Let  $M = \min(X_1, X_2, \dots, X_N)$  where each  $X_i \in [0, 1]$  is an uniform independent random variable

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

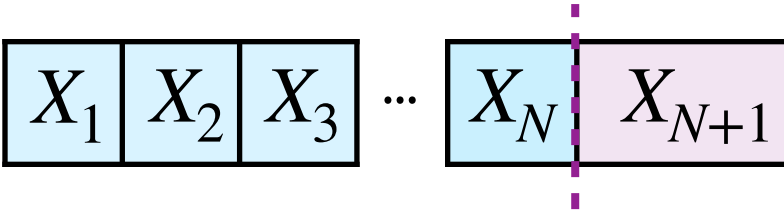
0

1



# Cardinality Sketch

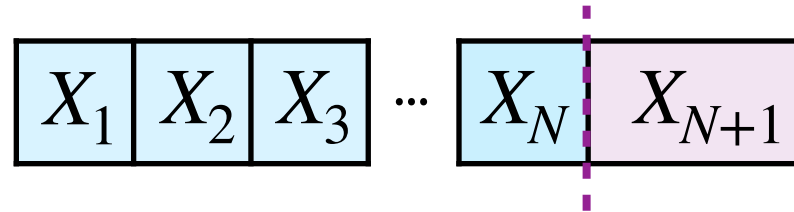
$\mathbf{E}[M]$  defines the range from 0 to the min value  $\left( M = \min_{1 \leq i \leq N} X_i \right)$

Consider an  $N + 1$  draw: 



# Cardinality Sketch

Consider an  $N + 1$  draw:



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

Define an **indicator**:

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

$\mathbf{E}[I_i] =$

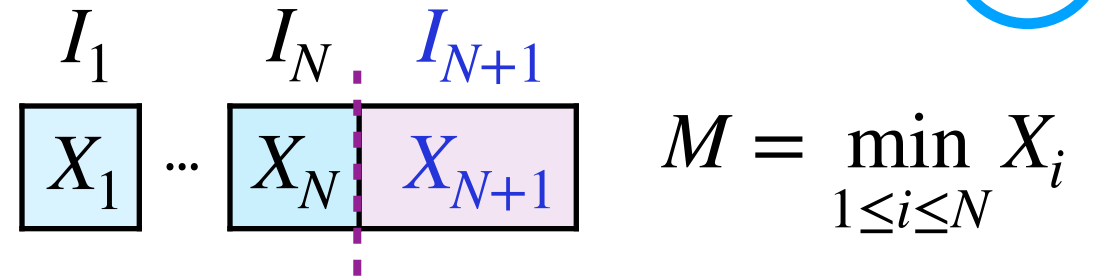


# Cardinality Sketch

*Hypothetical Draw*



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



By definition,  $\mathbf{E}[I_{N+1}] = \Pr(X_{N+1} < M) = \frac{1}{N+1}$



# Cardinality Sketch

**Claim:**  $E[M] = \frac{1}{N+1}$

$$N \approx \frac{1}{M} - 1$$

$\$ \#s \leftrightarrow$

$$\frac{1}{.328} - 1 \approx \underline{\underline{2}}$$

**Attempt 1**

0.962	0.328	0.771	0.952	0.923
-------	-------	-------	-------	-------

**Attempt 2**

0.253	0.839	0.327	0.655	0.491
-------	-------	-------	-------	-------

**Attempt 3**

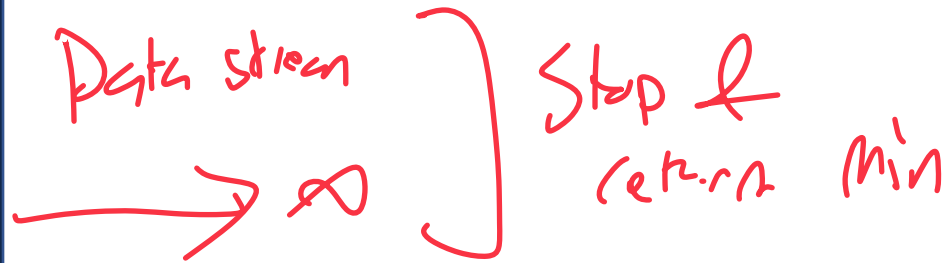
0.134	0.580	0.364	0.743	0.931
-------	-------	-------	-------	-------

$$\frac{1}{.134} - 1 \approx \underline{\underline{6.5}}$$

# Cardinality Sketch

The minimum hash is a valid sketch of a dataset but can we do better?

↳ Store min & max?



↳ Storing actual truth  
↳ why can't we do this?

↳ Hash table  
array size N

I 0-1 integer

0

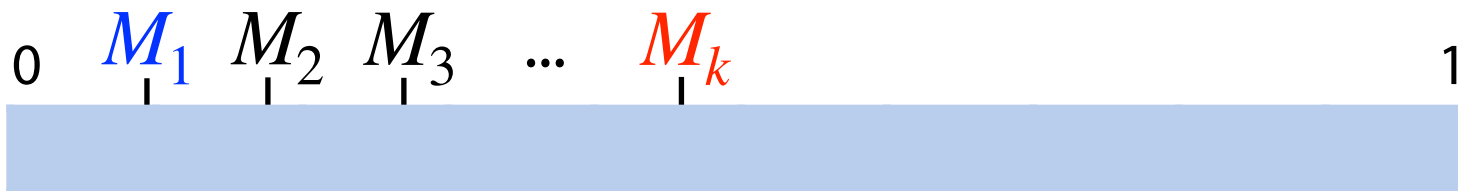
1



# Cardinality Sketch

**Claim:** Taking the  $k^{\text{th}}$ -smallest hash value is a better sketch!

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





# Cardinality Sketch

**Claim:** Taking the  $k^{\text{th}}$ -smallest hash value is a better sketch!

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

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

$M_1$   
|

$M_2$   
|

$M_3$   
|

...

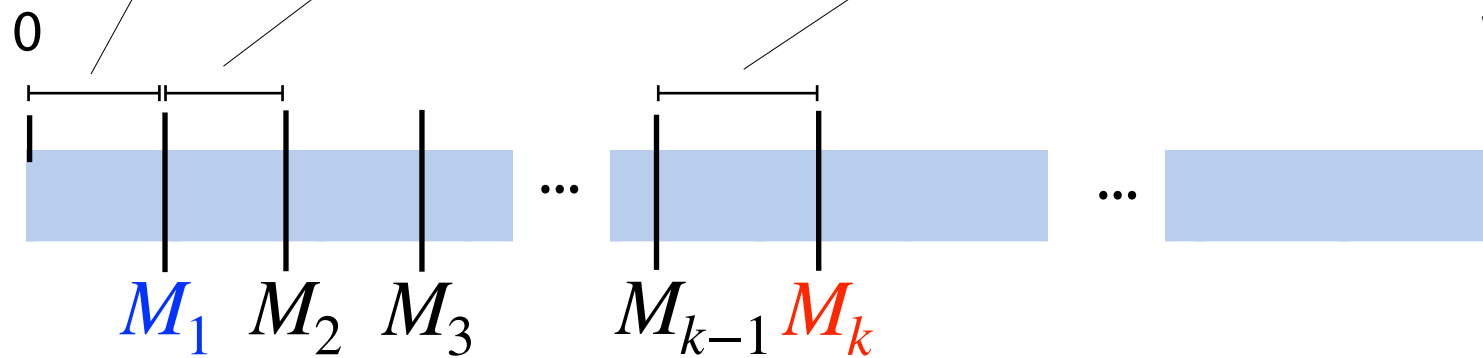
$M_{k-1}$   
|

$M_k$   
|

# Cardinality Sketch

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

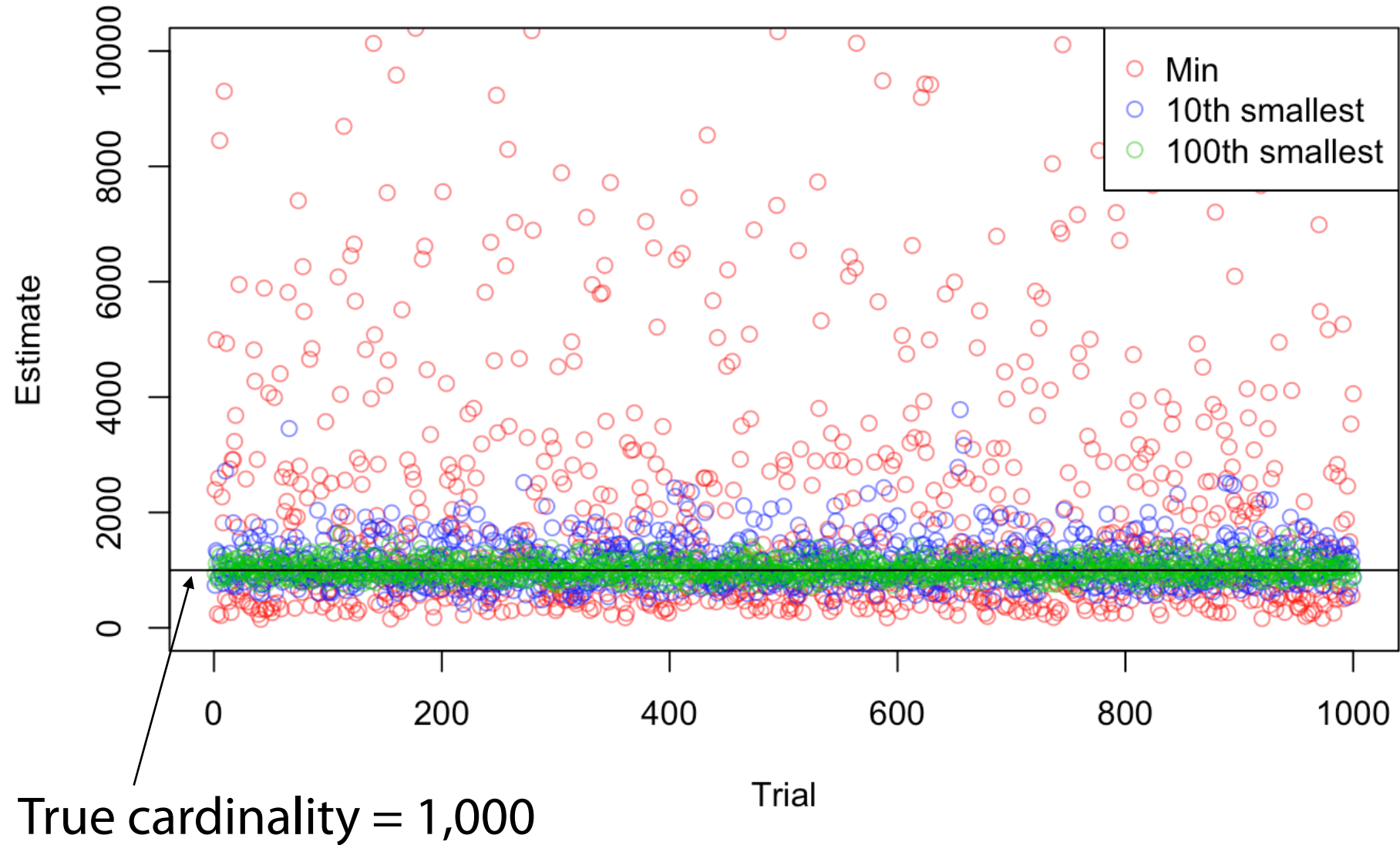
$$= \left[ \underbrace{\mathbf{E}[M_1]} + \underbrace{(\mathbf{E}[M_2] - \mathbf{E}[M_1])} + \dots + \underbrace{(\mathbf{E}[M_k] - \mathbf{E}[M_{k-1}])} \right] \cdot \frac{1}{k}$$



$k^{\text{th}}$  minimum  
value (KMV)

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

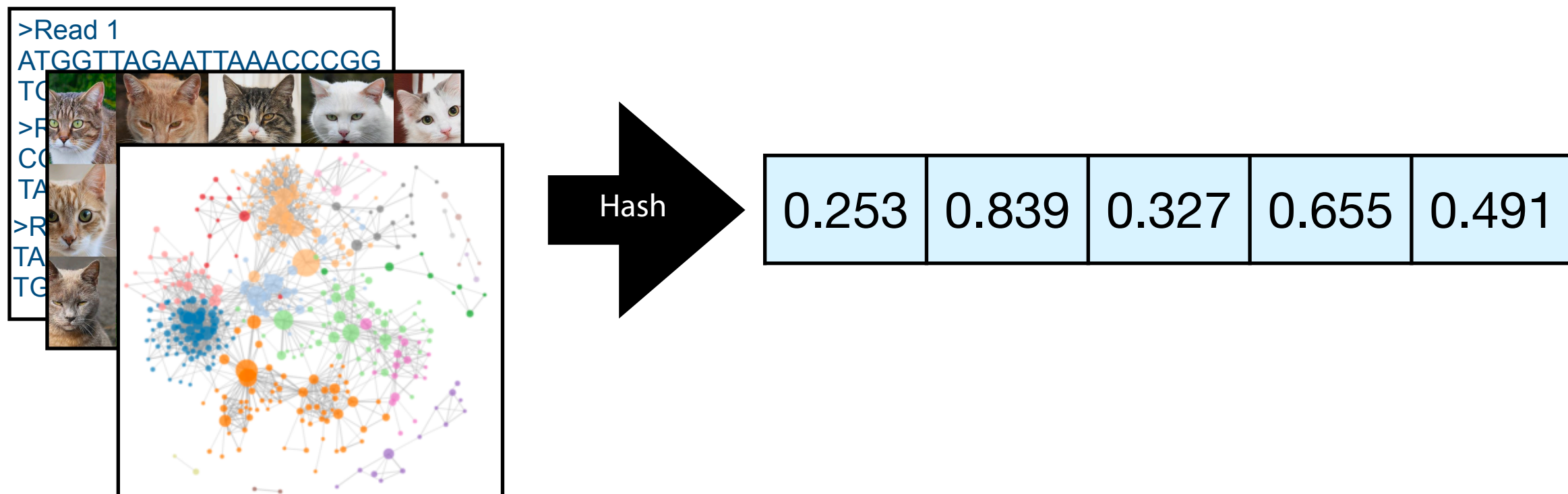
# Cardinality Sketch



# Cardinality Sketch



Given any dataset and a SUHA hash function, we can **estimate the number of unique items** by tracking the **k-th minimum hash value**.



To use the k-th min, we have to track k minima. **Can we use ALL minima?**

# Applied Cardinalities

Cardinalities

$$\frac{|A|}{|B|}$$

$$\frac{|A \cup B|}{|A \cap B|}$$

Set similarities

$$O = \frac{|A \cap B|}{\min(|A|, |B|)}$$

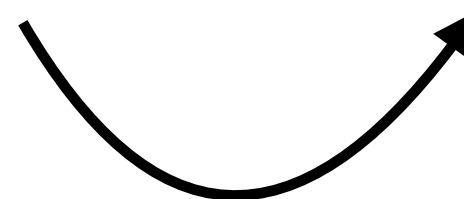
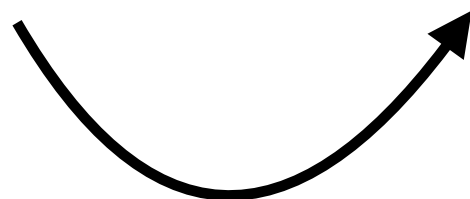
$$J = \frac{|A \cap B|}{|A \cup B|}$$

Real-world  
Meaning

AGGCCACAGTGTATTATGACTG  
||||| |||||  
AGGCCACAGTGAGTTATGACTG

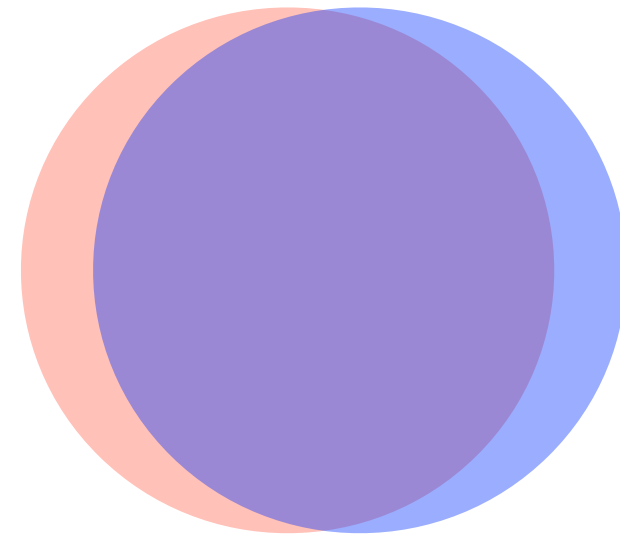
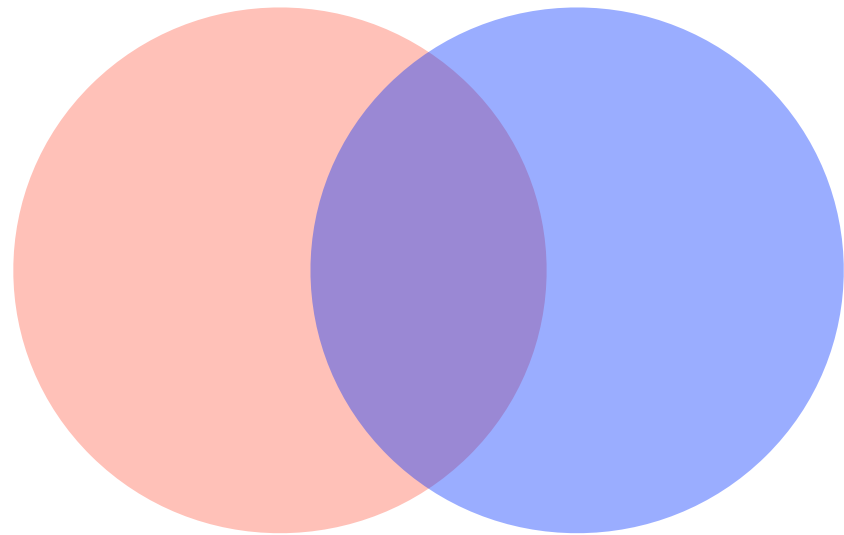
AAAAAAAAAAGATGT-AAGTA  
||||| |||||  
AAAAAAAAAAGATGTAAAGTA

GAGG--TCAGATTCACAGCCAC  
|||| |  
GAGGGGTCAGATTCACAGCCAC



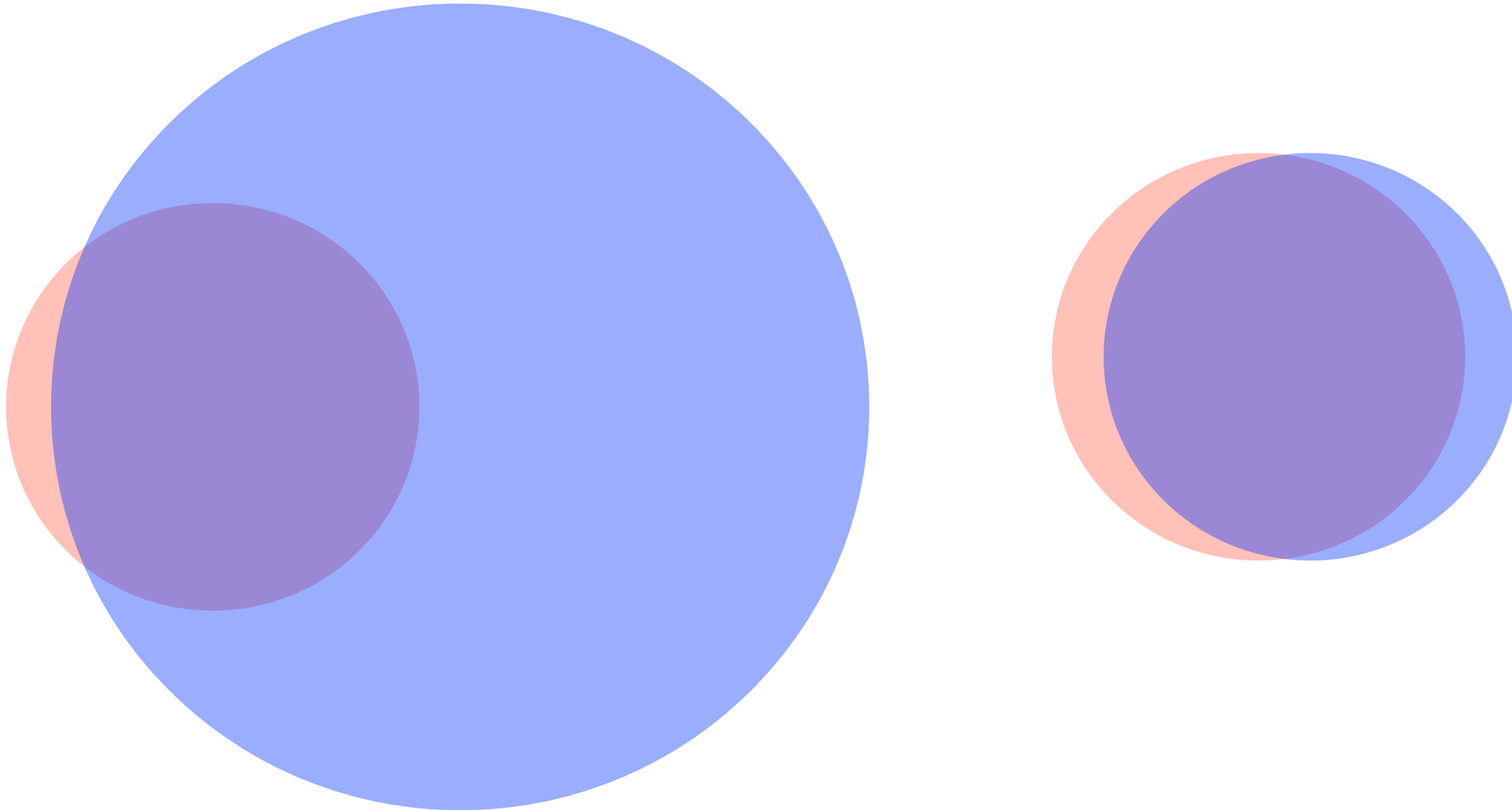
# Set Similarity Review

How can we describe how *similar* two sets are?



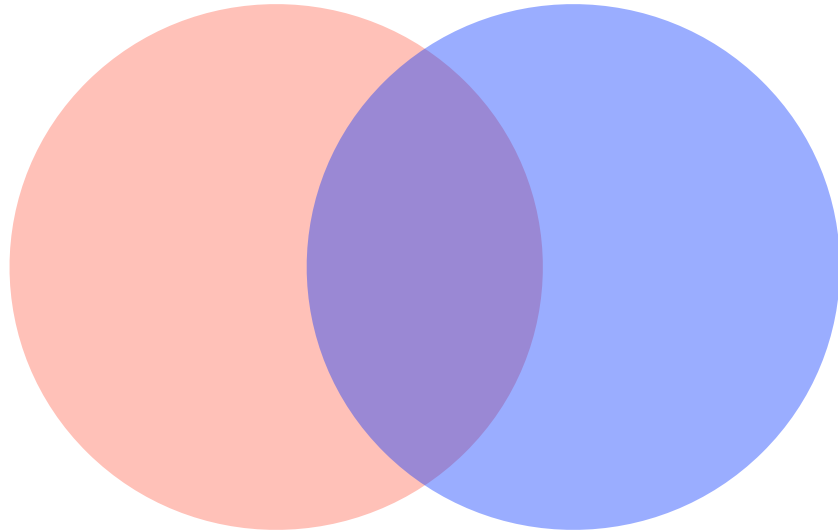
# Set Similarity Review

How can we describe how *similar* two sets are?



# Set Similarity Review

To measure **similarity** of  $A$  &  $B$ , we need both a measure of how similar the sets are but also the total size of both sets.

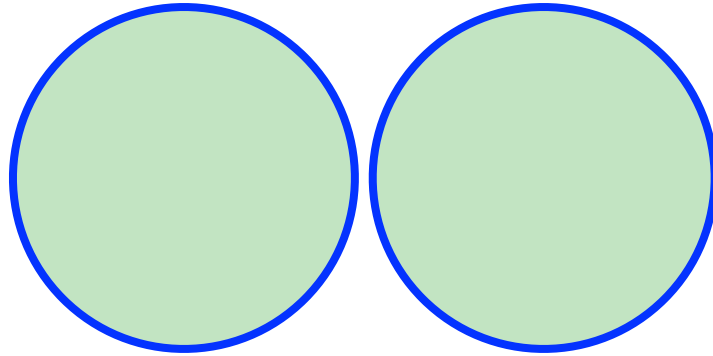


$$J = \frac{|A \cap B|}{|A \cup B|}$$

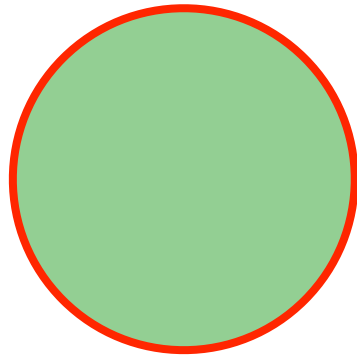
$J$  is the **Jaccard coefficient**



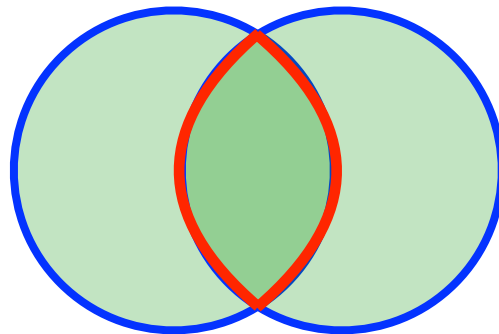
# Set Similarity Review



$$\frac{|A \cap B|}{|A \cup B|} = 0$$



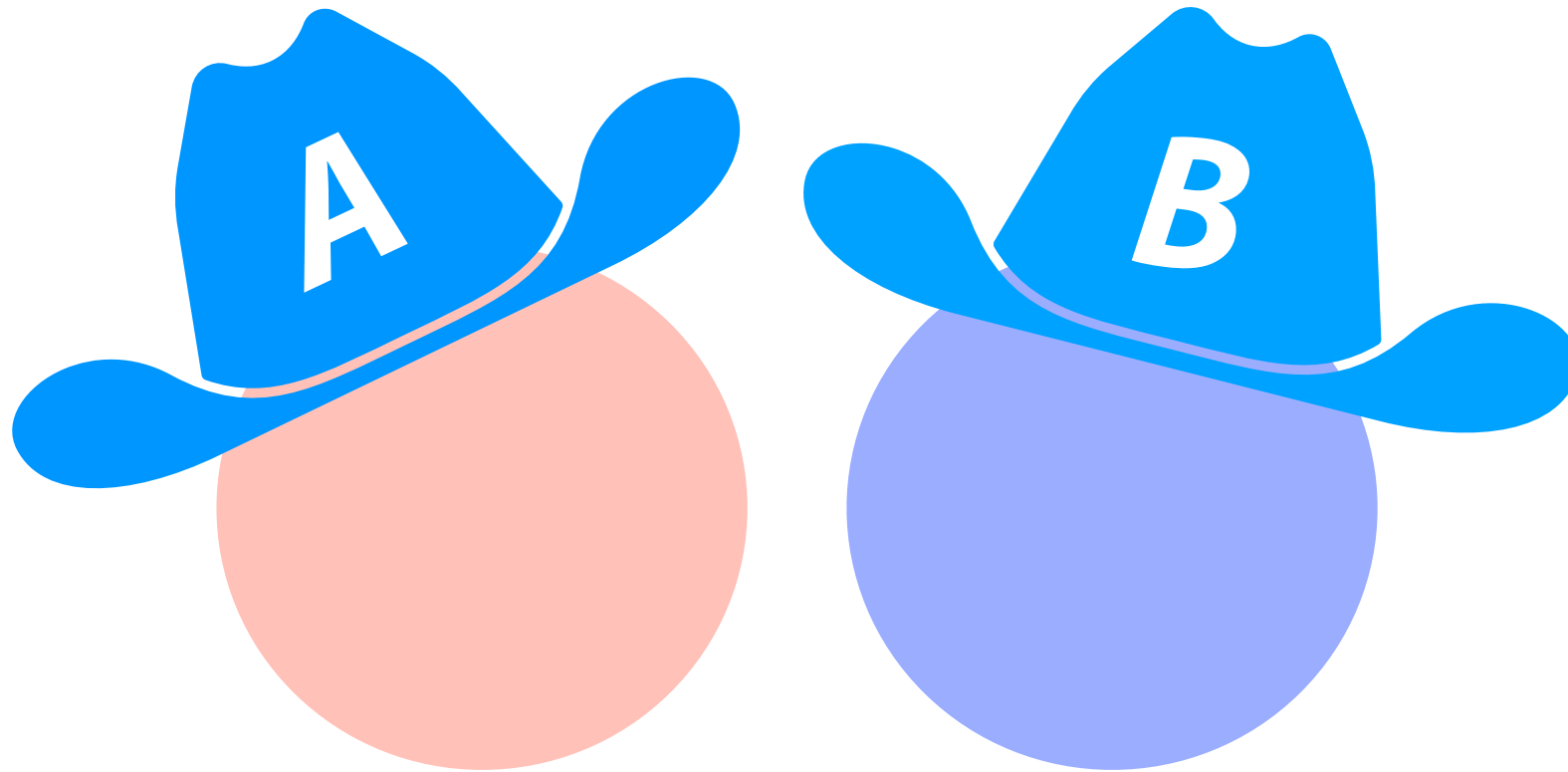
$$\frac{|A \cap B|}{|A \cup B|} = 1$$



$$0 < \frac{|A \cap B|}{|A \cup B|} < 1$$

# Similarity Sketches

But what do we do when we only have a sketch?



# Similarity Sketches

Imagine we have two datasets represented by their  $k$ th minimum values

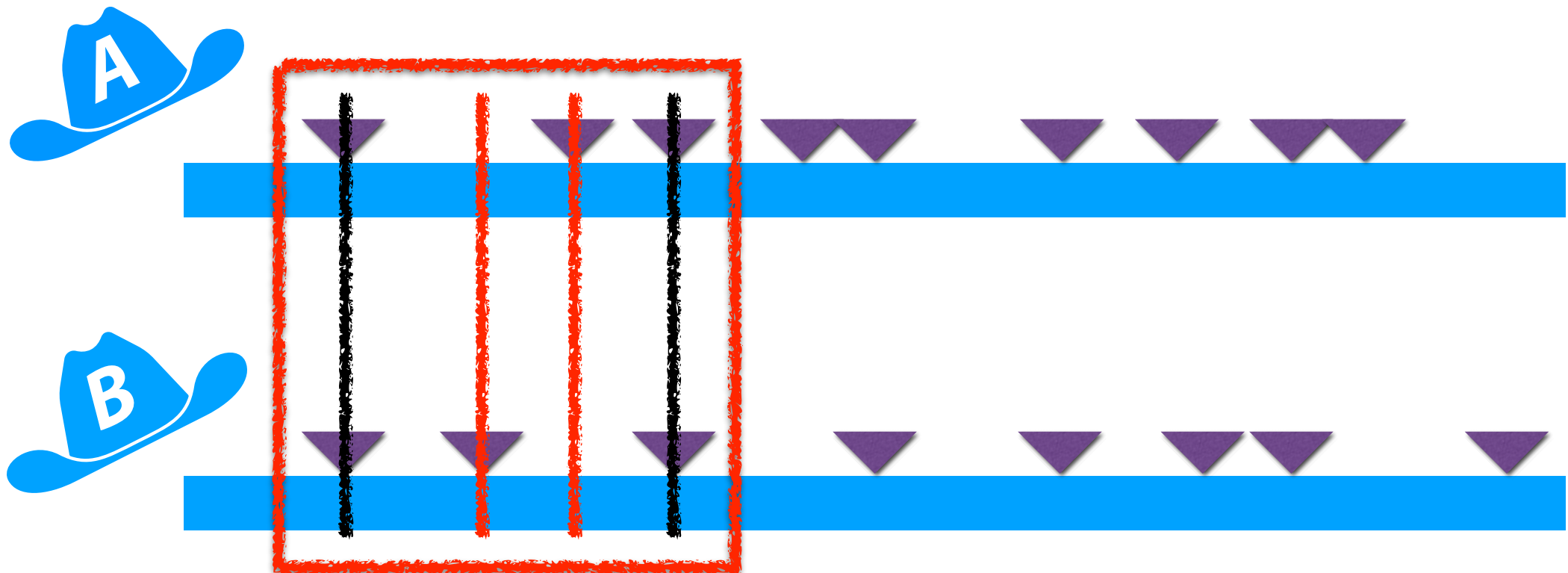


Image inspired by: Ondov B, Starrett G, Sappington A, Kostic A, Koren S, Buck CB, Phillippy AM. **Mash Screen: high-throughput sequence containment estimation for genome discovery.** *Genome Biol* 20, 232 (2019)

# Similarity Sketches

**Claim:** Under SUHA, set similarity can be estimated by sketch similarity!

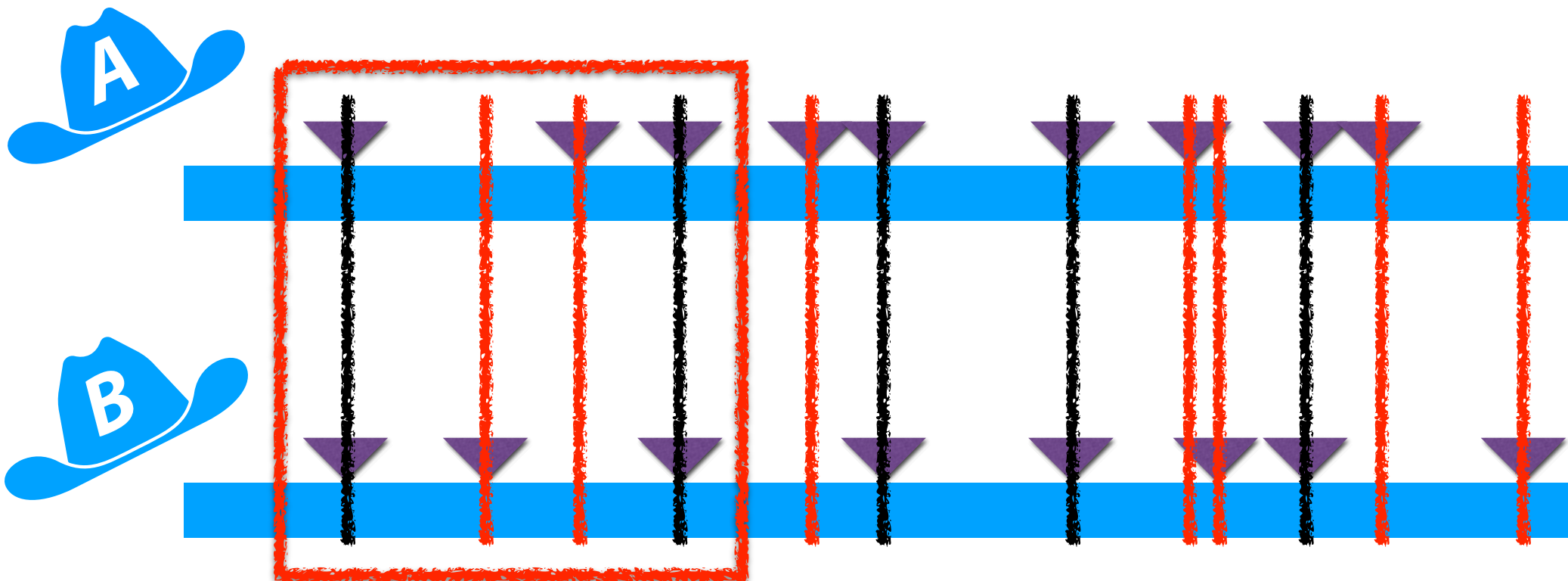
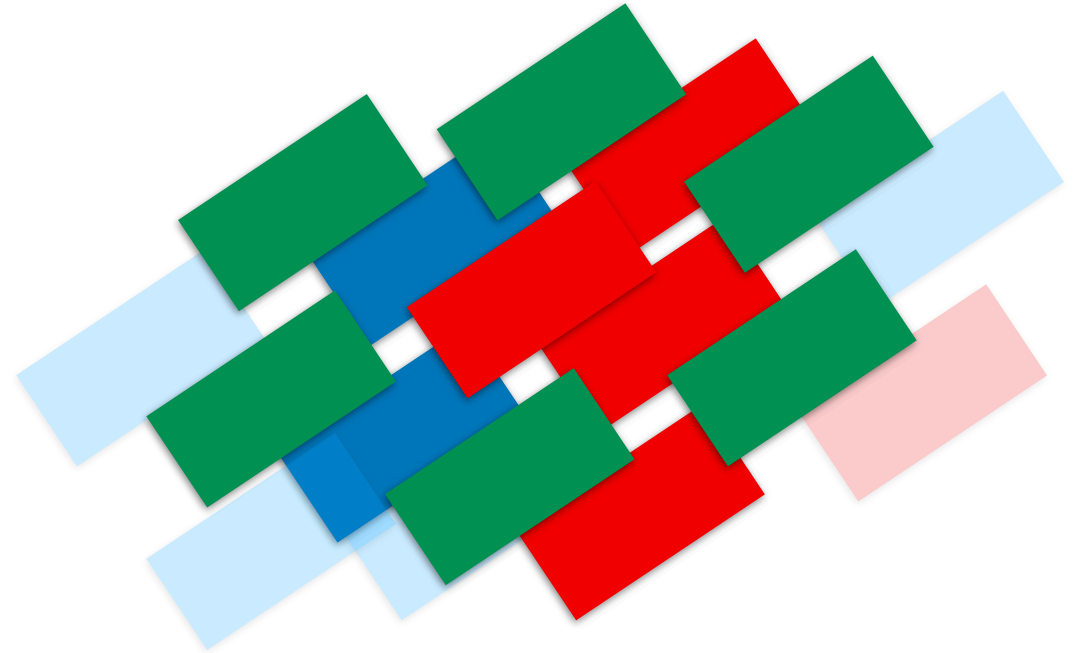
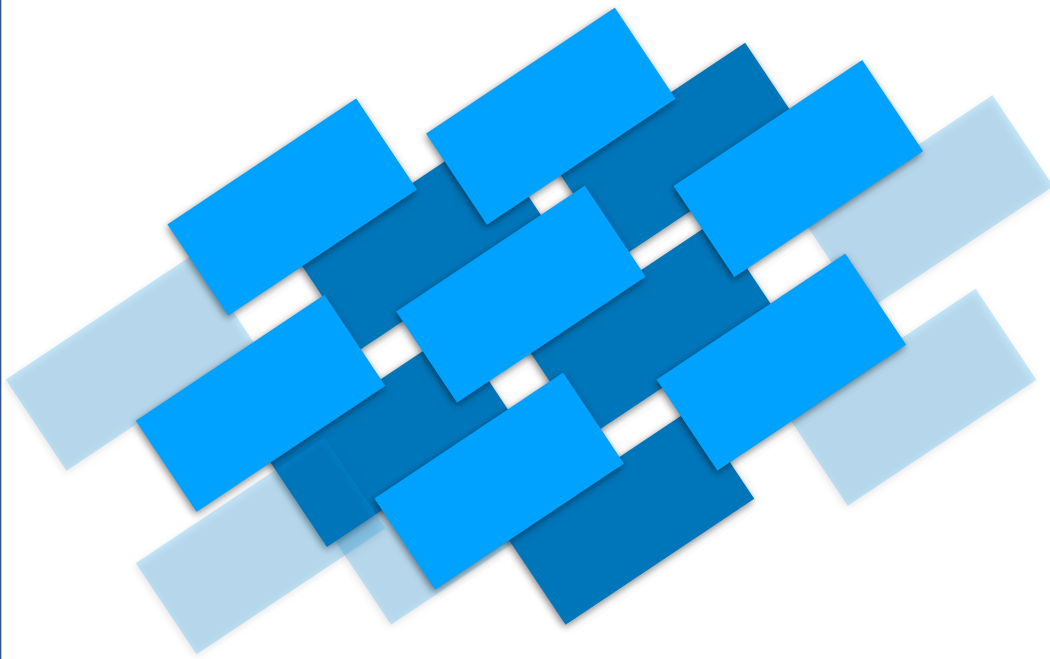


Image inspired by: Ondov B, Starrett G, Sappington A, Kostic A, Koren S, Buck CB, Phillippy AM. **Mash Screen: high-throughput sequence containment estimation for genome discovery.** *Genome Biol* 20, 232 (2019)

# Minhash Sketch

An approximation for a full dataset capable of **estimating set similarity**



# Minhash Sketch 'ADT' (Use Cases)

**Constructor**

**Cardinality Estimation**

**Set Similarity Estimation**

# MinHash Construction

A MinHash sketch has three required inputs:

- 1.

- 2.

- 3.

# MinHash Construction

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

$h(x) = x \% 7$

$k = 3$

