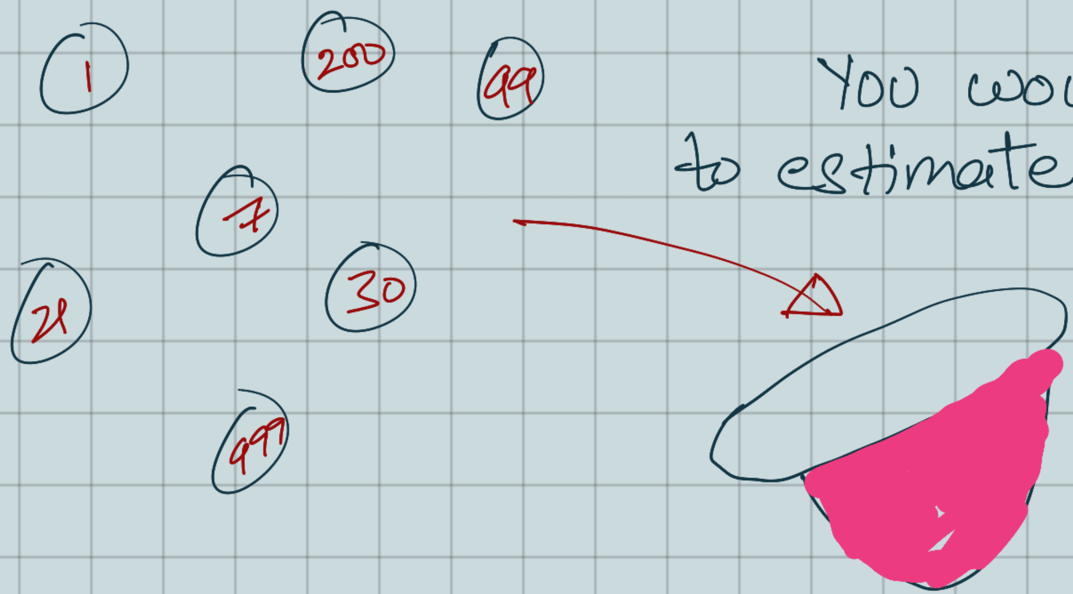
Hat problem:

- I take cards labeled 1--1000, 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 cards in the hat; what to pick?
  - median, minimum, maximum.

- What if the minimum was 500? 10? 4?

- Estimate should grow as minimum shrinks!

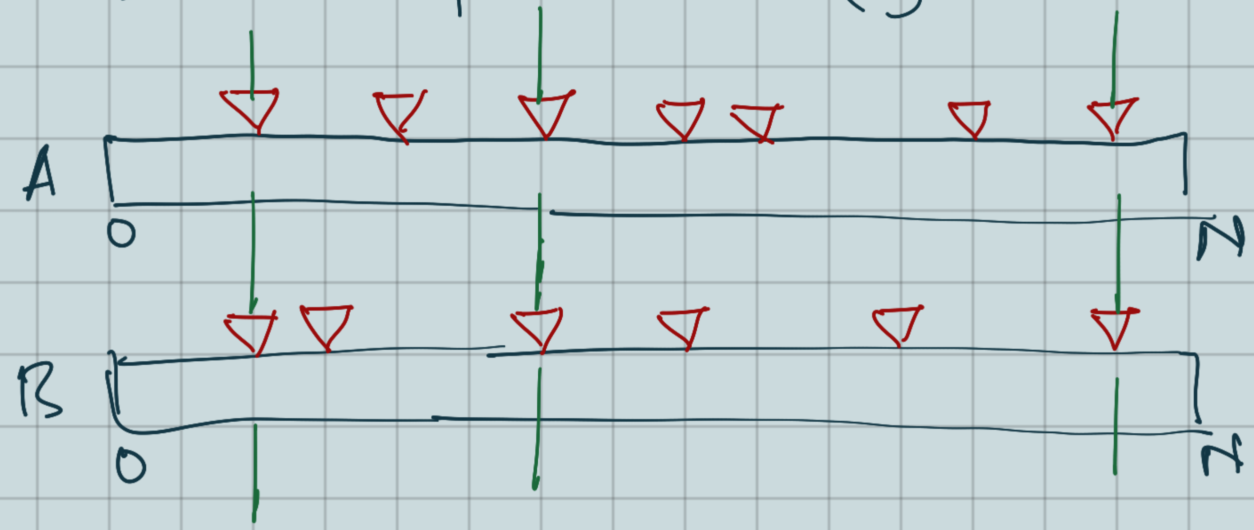
minimum $= 40$

$$40 \approx \frac{1000}{n+1}$$

$$n \approx 24$$

Easy to compute, fits in 10 bits

Two-hat problem: (functions on Sets)



→ Space of coincidences is large.

→ Need to look at more than one representative.

☆ → Instead of taking just the minimum, consider bottom k

→ we can estimate the cardinality of $A \cap B$, $A \cup B$

☆ → Instead of bottom-K, consider minimum in each of 3 partitions.

→ Accomplishes something similar to bottom-K.

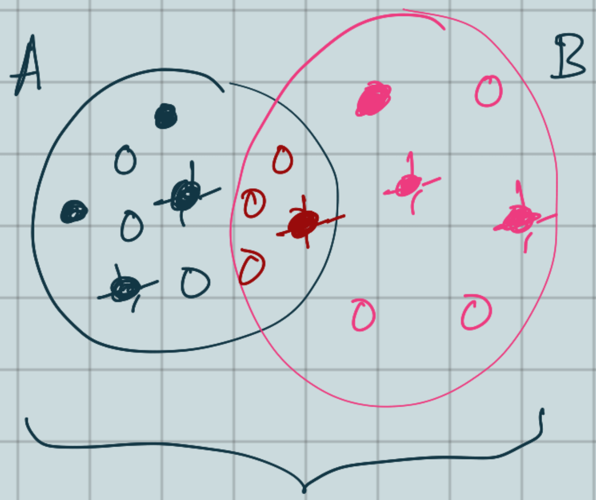
Mash: (Bottom-K Minhash Sketch)

Similarity search over genomic sequences

→ Different types of genomic sequences

- → Genomes
- → Metagenomes
- → amino acids

K-mers → length-K subsequence.



Sketch (A)	S(A ∪ B)	Sketch(B)
42	42	66
64	64	82
82	66	87
128	82	104
139	87	127

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \approx \frac{|S(A \cup B) \cap S(A) \cap S(B)|}{|S(A \cup B)|}$$

→ Because $S(A \cup B)$ is a random sample of $(A \cup B)$, the fraction of elements in $S(A \cup B)$ that are shared by both $S(A)$ and $S(B)$ is an unbiased estimate of $J(A, B)$

Example :- (Cardinality)

A : {3, 7, 8, 11, 15, 17, 22, 23}

B : {2, 3, 6, 7, 9, 11, 17, 23}

Q: can we localize hash values in a venn diagram? Assume no collisions!

lets say $K = 8$.

$S(A \cup B) = \{2, 3, 6, 7, 8, 9, 11, 15\}$

J: fraction of items in union sketch that are in both: $\frac{3}{8} = 0.375$

Hyper Log Log :- (Cardinality)

Simple solution: hash table $\Omega(N)$ space

→ we are going to use randomness.

Hash function:

Bob → $\boxed{\text{hash}}$ → $\boxed{0101110011}$

$Pr(0) \rightarrow \frac{1}{2}$

$Pr(1) \rightarrow \frac{1}{2}$

→ Unique random binary strings

→ Coin game : Flip a coin!

→ If H, flip again.

→ If T, stop!

H T

H H T

T

H T

→ Probability of getting exactly 3 Hs.

$$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \boxed{\frac{1}{16}}$$

→ Every 16 attempts one sequence will have exactly 3 Hs.

→ Another way of saying:-

If we see a sequence of 3Hs, there are probably 16 items.

→ If in all hash values, the longest streak of H is L

→ then average 2^{L+1} items.

→ Count leading 0s in hashes.

→ Estimate the total number of distinct items.

→ Leading 0s = 2

Cardinality Estimate $= 2^{2+1} = 8$

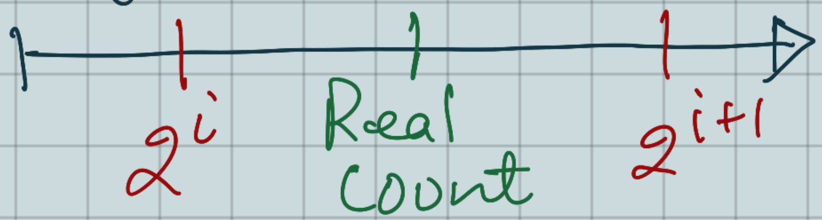
M = Maximum number of unique elements

$$2^{L+1} \leq M \Rightarrow L \leq \lg M.$$

Bit length of L = $\boxed{\lg L = \lg \lg M.}$

Problems: $\longrightarrow$ $\boxed{A \cdot C}$

→ Will only estimate power of 2



→ Too much luck involved.

Solution:

Use multiple approximate counters and average results.

$$\underbrace{\boxed{A\cdot C}\ \boxed{A\cdot C}\ \boxed{A\cdot C}\ \boxed{A\cdot C}\ \cdots\ \boxed{A\cdot C}} \xrightarrow{\text{Average}} \boxed{\text{Final Result}}$$

$L_1 \quad L_2 \quad L_3 \quad L_3 \quad L_K$

$$2^{\left(\frac{L_1 + L_2 \cdots L_K}{K}\right)}$$

→ Can have bias.

Hyperloglog :-

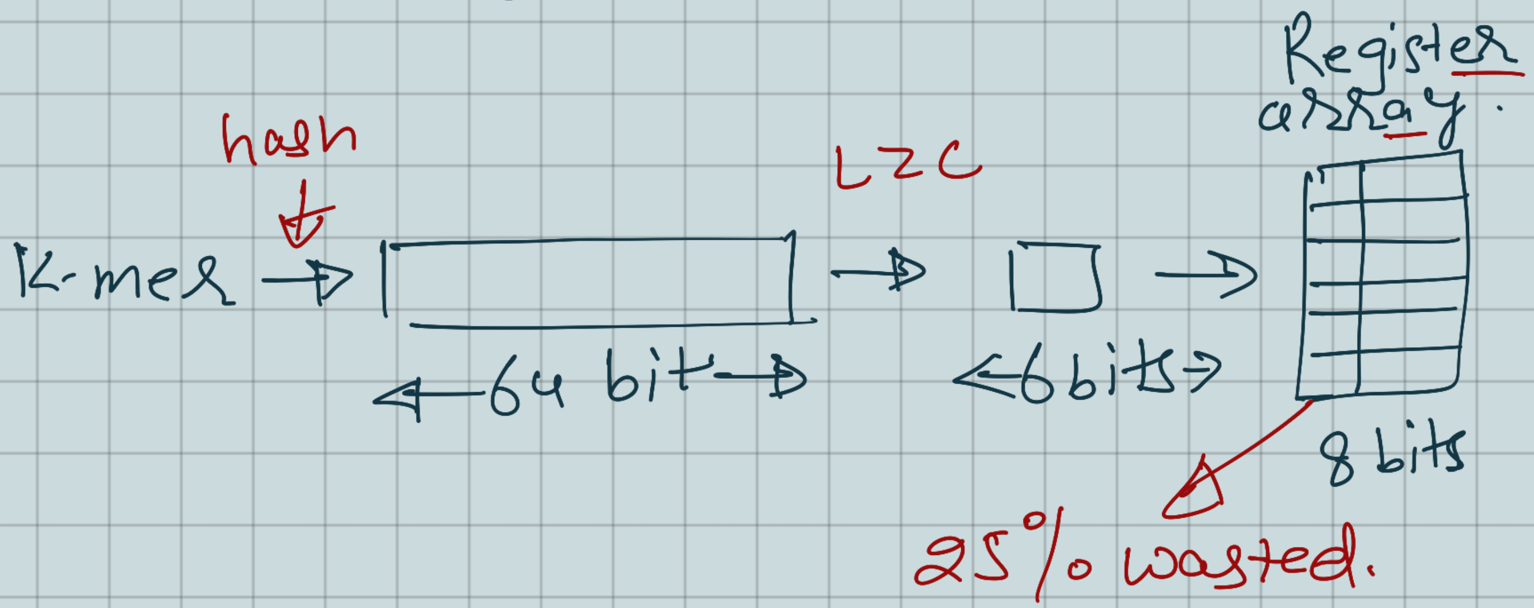
Harmonic mean of K counters.

$$\frac{K}{\frac{1}{L_1} + \frac{1}{L_2} + \cdots \frac{1}{L_K}}$$

→ less sensitive to large outliers.

Dashing :- HyperLogLog.

1. K-partition
2. $\lfloor \lg_2 N \rfloor$
3. Re-exponetiation.
4. Averaging bias correction.



→ Instead of LZC, use truncated log

$\lfloor \lg_{1.19} \rfloor \rightarrow$ 8 bits.

→ Better cardinality estimate