Week 2 Tuesday

- Self-grade due Friday
- Thoughts on "working smart not hard"
  - ↳ focus is a muscle
    (fast dopamine weakens it)
  - ↳ even 15min is enough to start a hard task

Plan:

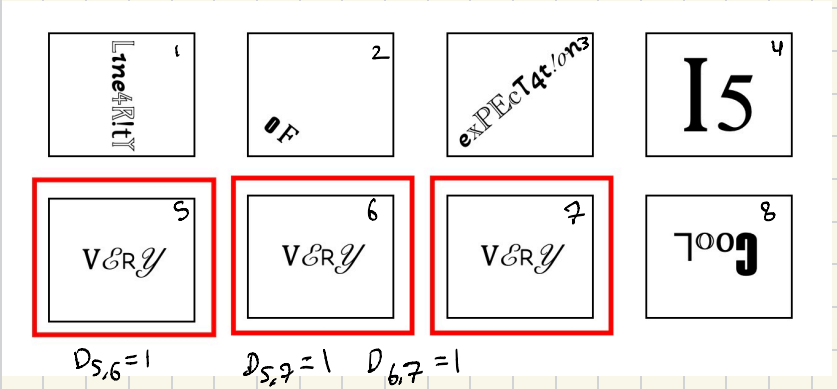
- Finish up set size estimation
  (Formalize "unlikely" event?)
- Start frequent items

Problem: Given query access, how big is a set?

$$D_{i,j} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ same} \\ 0 & \text{else} \end{cases}$$

$$D = \sum_{i,j=1 \,;\, i<j}^{m} D_{i,j}$$

$$\mathbb{E}[D] = \sum_{i<j} \mathbb{E}[D_{i,j}] = \binom{m}{2} \frac{1}{n} = \frac{m(m-1)}{2n}$$



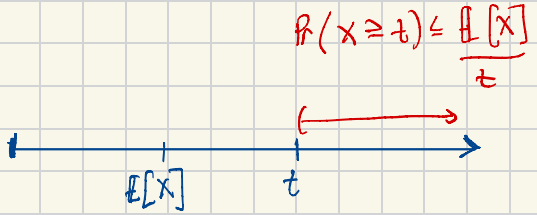
Suppose we made $m = 1{,}000$ queries and saw $D = 10$ duplicates.

How does this compare to what we expect?

Markov's Inequality

Theorem: For any non-negative rv X and $t > 0$,

$$\Pr(X \geq t) \leq \frac{\mathbb{E}[X]}{t}$$



Proof:

$$\begin{aligned}
\mathbb{E}[X] &= \sum_x x \Pr(X = x) \\
&= \sum_{x:\, x \geq t} x \Pr(X = x) + \sum_{x:\, x < t} x \Pr(X = x) \\
&\geq \sum_{x:\, x \geq t} t \Pr(X = x) + 0 = t \Pr(X \geq t)
\end{aligned}$$

Answer to duplicate question:

Frequent Items

Problem: Most common elements in a stream?

Examples:
- most popular products on Amazon
- most watched videos on YouTube
- most searched queries on Google

A stream of n items $x_1, \dots, x_n$

U = set of all items

K positive integer and $\epsilon > 0$ small constant

Return:
- every item that appears at least $\frac{n}{K}$ times
- only items that appear at least $(1-\epsilon)\frac{n}{K}$ times

Naive Attempt: Store each item with its frequency

Is this a good solution?

Let $f(v) = \sum_{i=1}^{n} \mathbb{1}[x_i = v]$

↑ frequency of item v

Our strategy:

Maintain estimates $\hat{f}(v)$ s.t.

$$f(v) \leq \hat{f}(v) \leq f(v) + \frac{\epsilon}{k} n$$

Question: How do we use $\hat{f}$ to solve frequent items problem?

Hash Functions

Definition: $h: U \to \{1, \ldots, m\}$ uniform random if

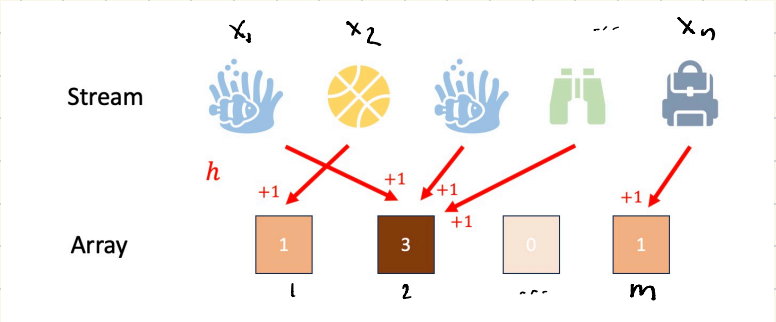
- $\Pr(h(x) = i) = \frac{1}{m}$ for all x, i
- $h(x)$ and $h(y)$ are independent for all x, y

$\Rightarrow \Pr(h(x) = h(y)) = \ldots$

Note: Cannot efficiently implement uniform random, but universal suffices:

$$\Pr(h(x) = h(y)) \leq \frac{1}{m}$$

Counting



For x_i in $x_1, \dots, x_n$:

$$A[h(x_i)] \leftarrow A[h(x_i)] + 1$$

Return $\hat{f}(v) = A[h(v)]$

$$\hat{f}(v) = f(v) + \sum_{y \in U \setminus \{v\}} f(y)\, \mathbb{1}[h(y) = h(v)]$$

$f(v)$: true; $\sum_{y \in U \setminus \{v\}} f(y)\, \mathbb{1}[h(y) = h(v)]$: error

Show that:

$$\Pr\left(\sum_{y \in U \setminus \{v\}} f(y)\, \mathbb{1}[h(y) = h(v)] \geq \frac{2n}{m} \right) \leq \frac{1}{2}$$

With one array,

$$f(v) \le A[h(v)] \le f(v) + \frac{2n}{m}$$

wp $\frac{1}{2}$ for any v

Setting $m = \frac{2k}{\epsilon}$,

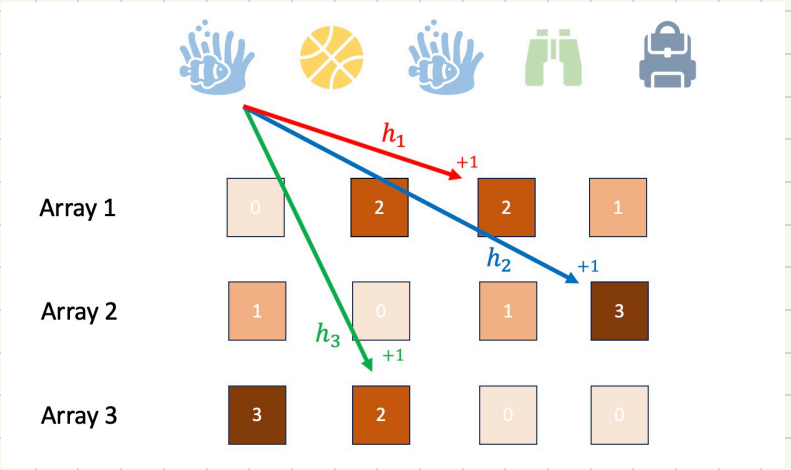
$$f(v) \le A[h(v)] \le f(v) + \frac{n}{k}\epsilon$$

wp 1/2 for any v

↑ a little low...

Strategy: Boost success by repeating subroutine!

Count-Min



For x_i in $x_1, \ldots, x_n$:

For A_j in $A_1, \ldots, A_t$:

$$A_j[h_j(x_i)] \leftarrow A[h_j(x_i)] + 1$$

Return $\hat{f}(v) = \min_j A_j[h_j(v)]$