

2025-02-11:

- Wrapping up  $k$ -wise independent hashing (see the previous note)
- Estimating the number of distinct elements

2025-02-12:

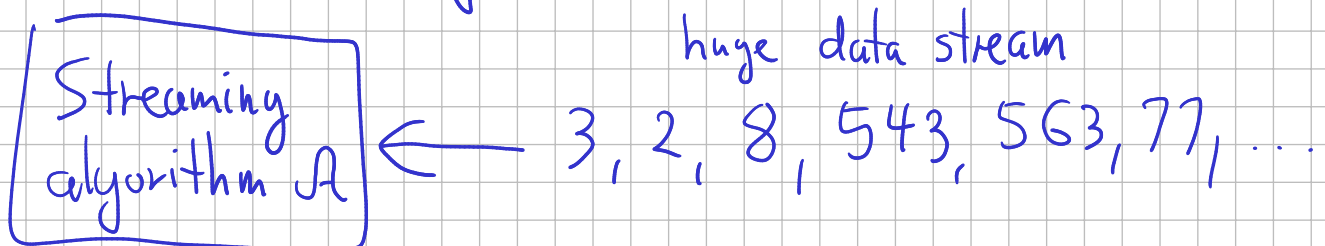
- Estimating the number of distinct elements

2025-02-13:

- A few more comments on estimating the number of distinct elements
- Space complexity of estimating  $F_p$
- Graph connectivity sketches (see the next note)

---

Model: Streaming algorithms



$X$  - universe of elements of the stream

for  $x \in X$ ,  $f(x) = \#$  occurrences of  $x$

Goal:

Estimate number of distinct elements

$$F_0 = |\{x \in X : f(x) \neq 0\}|$$

More precisely: compute  $(1+\epsilon)$ -multiplicative approximation

$$(1-\epsilon) F_0 \leq \hat{F}_0 \leq (1+\epsilon) F_0$$
$$\Leftrightarrow |\hat{F}_0 - F_0| \leq \epsilon F_0$$

---

Algorithm (insertions only)

$h =$  "fully random" hash function from  $X$  to  $\mathbb{N}$  s.t.

$$\Pr(h(x) = i) = 2^{-(i+1)}$$

$h(x) =$	0	1	2	...
$\Pr(\quad) =$	1/2	1/4	1/8	...

Initialization:

$$z \leftarrow 0$$

$$A \leftarrow \emptyset$$

Processing:

For each element  $x$  in the stream:

if  $h(x) \geq z$ :

$$A \leftarrow A \cup \{(x, h(x))\}$$

$$\text{while } |A| \geq \frac{C}{\epsilon^2}$$

$$z \leftarrow z + 1$$

remove from  $A$  all pairs  $(y, g)$  s.t.  $g < z$

Set-theoretic union,  
so if  $(x, h(x))$  already  
in  $A$ ,  $A$  won't change.  
 $\{(1,2)\} \cup \{(1,2)\} = \{(1,2)\}$

$C$  - sufficiently  
large constant  
( $C = 576$ )

Estimate: Output  $|A| \cdot 2^z$

Intuition:

- initially, we try to store all elements in  $A$
- but if there are too many, we drop half in expectation and store only  $x$ 's with  $h(x) \geq 1$  (threshold  $z$  becomes 1)
- if this is also too many, increase  $z$  further and drop half of elements in expectation again & repeat as needed
- multiply the size of the final  $A$  by 2 to the number of times it was "halved" in expectation

# Analysis:

Random variables:

$Z$  - final value of  $Z$

$$X_{i,x} = \begin{cases} 1 & \text{if } h(x) \geq i \\ 0 & \text{o.w.} \end{cases} \quad (\text{for } x \in X, i \in \mathbb{N})$$

$$Y_i = \sum_{x: f(x) > 0} X_{i,x} \quad (\text{for } i \in \mathbb{N})$$

Output of the algorithm:  $Y_Z \cdot 2^Z$

Incorrect output:

$$|Y_Z \cdot 2^Z - F_0| > \varepsilon F_0$$

$$\Leftrightarrow \left| Y_Z - \frac{F_0}{2^Z} \right| > \frac{\varepsilon F_0}{2^Z}$$

Observation: if  $Z=0$ ,  $|A| = Y_0 = F_0$

The algorithm outputs  
exact value

$$\underbrace{\Pr(\text{incorrect output})}_{= \otimes} = \sum_{i=1}^{\infty} \Pr\left(|Y_i - \frac{F_0}{2^i}| > \frac{\varepsilon F_0}{2^i} \wedge Z=i\right)$$

Select integer  $s$  s.t.

$$\frac{12}{\varepsilon^2} \leq \frac{F_0}{2^s} < \frac{24}{\varepsilon^2}$$

Observation: if  $s \leq 1$ ,  $F_0 < \frac{48}{\varepsilon^2} \ll \frac{C}{\varepsilon^2}$

Hence  $Z=0$  and the algorithm outputs the exact value

So assume  $s > 1$  for the rest of the proof

$$\otimes \leq \underbrace{\sum_{i=1}^{s-1} \Pr\left(|Y_i - \frac{F_0}{2^i}| > \frac{\varepsilon F_0}{2^i}\right)}_{\triangle} + \underbrace{\sum_{i=s}^{\infty} \Pr(Z=i)}_{\square}$$

$$\square = \Pr(Z \geq s) = \Pr(Y_{s-1} \geq C/\varepsilon^2)$$

$$\stackrel{\text{Markov's}}{\leq} \frac{\mathbb{E}[Y_{s-1}]}{C/\varepsilon^2} = \frac{F_0/2^{s-1}}{C/\varepsilon^2} = \frac{2\varepsilon^2}{C} \cdot \frac{F_0}{2^s}$$

$$\stackrel{\text{Selection of } s}{\leq} \frac{2\varepsilon^2}{C} \cdot \frac{24}{\varepsilon^2} = \frac{48}{C} < \frac{1}{12} \quad \uparrow \text{large } C$$

$$\Delta \leq ?$$

$$\mathbb{E}[Y_i] = \frac{F_0}{2^i}$$

$$\text{Var}[Y_i] = \sum_{x: f(x) > 0} \text{Var}[X_{i,x}] \leq \sum_{x: f(x) > 0} \mathbb{E}[X_{i,x}^2]$$

$$= \sum_{x: f(x) > 0} \mathbb{E}[X_{i,x}] = \mathbb{E}[Y_i] = \frac{F_0}{2^i}$$

$$\Delta = \sum_{i=1}^{s-1} \Pr\left(|Y_i - \mathbb{E}[Y_i]| > \frac{\varepsilon F_0}{2^i}\right)$$

$$\leq \sum_{i=1}^{s-1} \Pr\left(|Y_i - \mathbb{E}[Y_i]| > \varepsilon \sqrt{\frac{F_0}{2^i}} \cdot \sqrt{\text{Var}[Y_i]}\right)$$

$$\leq \sum_{i=1}^{s-1} \frac{2^i}{\varepsilon^2 F_0} = \frac{1}{\varepsilon^2 F_0} \sum_{i=1}^{s-1} 2^i \leq \frac{2^s}{\varepsilon^2 F_0} \leq \frac{1}{\varepsilon^2} \cdot \frac{\varepsilon^2}{12} = \frac{1}{12}$$

Chebyshev's inequality

selection of  $s$

$$\textcircled{*} \leq \Delta + \square \leq \frac{1}{12} + \frac{1}{12} = \frac{1}{6}$$

Conclusion: the algorithm outputs a  $(1+\varepsilon)$ -multiplicative approximation w.p. at least  $5/6$

Space usage:

$O(1/\epsilon^2)$  elements of  $X$  +  $O(1/\epsilon^2)$  integers

can be reduced by hashing  
into a small range  
(2-wise independence  
is sufficient)

at most  $O(\log \log m)$   
bits with high probability  
for further space savings  
( $m = \text{stream length}$ )

Optimal:  $O(1/\epsilon^2 + \log n)$  bits for  $X = [n]$

Hash function  $h$ ?

- 2-wise independence suffices, needed for  $\text{Var}[Y_i]$
- Non-uniform distribution on some range?  
See Homework 3?

---

Practical  $F_0$  algorithms? See:

- HyperLogLog
- HyperLogLog++

---

Space complexity of streaming  $F_p$   
 $X = [n]$ , stream length =  $O(\text{poly}(n))$

$p \in [0, 2]$ :  $O(\frac{1}{\epsilon^2} \log n)$

$p > 2$ :  $n^{1-2/p} \cdot \text{poly}\left(\frac{\log n}{\epsilon}\right)$