- Recap
- Fingerprinting Analysis
- Bloom Filter

Recap

Problem: Set Membership

Store set of m elements $S = \{s_1, \ldots, s_m\}$
- each element $s_i \in U$ ← U: large set "the universe"

Goal: Answer "Is $x \in S$?" queries
- want to minimize space
- can't have false Negatives but OK $\Pr[\text{false positive}] \leq 1/10$

Last week solution: Fingerprinting

Let $F : U \to \{0,1\}^b$ be random hash function

Store $F(s_1), F(s_2), \ldots, F(s_m)$ in sorted list of b-bit fingerprints

Query(x):
(1) $z \leftarrow F(x)$
(2) search for z in list
(3) return YES iff z found

Analysis

space: bm bits

time: $b\log(m)$

$\log(m)$ search

b compare two b-bit strings

error:

- if $x \in S$ then always output YES ✓
- if $x \notin S$???

Error Analysis when $x \notin S$:

↳ F is random, so for each $s_i \neq x$ $\Pr[F(s_i) = F(x)] = \frac{1}{2^b}$

$\Pr[\text{False Positive}] = \Pr[\exists i \text{ s.t. } F(s_i) = F(x)]$

$= 1 - \underbrace{\left(1 - \frac{1}{2^b}\right)}_{\Pr[F(s_i) \neq F(x)]}{}^{m} \leftarrow m$ possible matches

We want $\Pr[\text{False Positive}] \leq \frac{1}{10}$

Recall: – $1 + x \leq e^x$

– for $0 \leq x \leq \frac{1}{2}$ $1 - x \geq e^{-2x}$

$1 - \frac{1}{2^b} \geq e^{-\frac{2}{2^b}} \Rightarrow \left(1 - \frac{1}{2^b}\right)^m \geq e^{-\frac{2m}{2^b}} \geq 1 - \frac{2m}{2^b}$

$\Rightarrow$ $\Pr[\text{False Positive}] = 1 - \left(1 - \frac{1}{2^b}\right)^m$

$\leq 1 - \left(1 - \frac{2m}{2^b}\right)$

$= \frac{2m}{2^b}$ if $\frac{2m}{2^b} \leq \frac{1}{10} \Leftrightarrow$

$\leq \frac{1}{10}$ $2^b \geq 20m \Leftrightarrow \boxed{b = \log(20m)}$

Note: if $b = 2\log m$ then $\Pr[\text{error}] \leq \frac{2m}{2^b} = \frac{2m}{2^{2\lg m}} = \frac{2m}{m^2} = O\left(\frac{1}{m}\right)$

Take-home message: with $O(m \log m)$ space

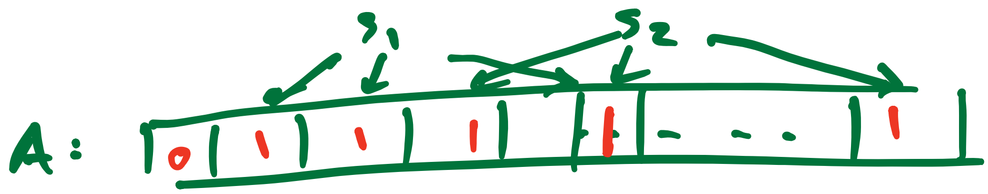
can support set membership in $O(\log m)$ time

false positive rate $\leq O\left(\frac{1}{m}\right)$

Bloom Filters

Our last data structure is also for set membership problem, allows for more nuanced time/space/error tradeoffs

Idea: Store data in n-bit array:



Let $F_1, F_2, \ldots, F_k : U \longrightarrow \{0, \ldots, n-1\}$ be k independent random hash functions

Store $s_1, \ldots, s_m$:

1. initialize $A[i] = 0$ for all $0 \leq i \leq n-1$
2. For each $1 \leq j \leq m$:
   Compute $z_\ell = F_\ell(s_j)$ for each hash function F_ℓ
   Set $A[z_\ell] = 1$

Query (x):
  for each hash function F_ℓ
    $z \leftarrow F_\ell(x)$
    if $A[z] = 0$ return NO
  return YES

Analysis:

space: n bits

time: $O(k)$

error:
if $x \in S$ then always output YES
if $x \notin S$???

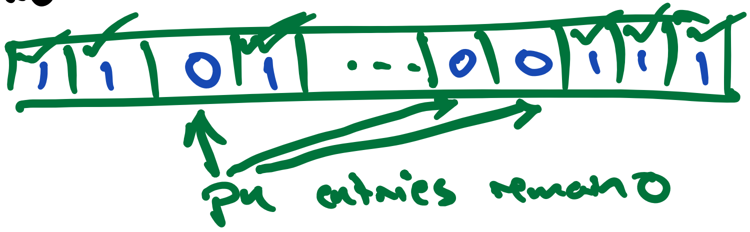
False positive rate: fix $x \notin S$

We hash m items from S, each using k functions

$\Rightarrow$ $m \cdot k$ balls into n bins

$$\Pr[\text{one bin remains empty}] = \left(1-\frac{1}{n}\right)^{mk} \approx e^{-\frac{mk}{n}} =: p$$

Now let's assume a p-fraction of entries in A are unoccupied



$$\Pr[\text{false positive}] = \Pr[\text{all } k \text{ hashes map to 1-values}]$$
$$= (1-p)^k$$
$$= \left(1 - e^{-\frac{mk}{n}}\right)^k \qquad (*)$$

The 2 factors of k are competing here.

more k: more chances to find $A[i]=0$ $\Rightarrow$ avoid false positive

less k: less constraints: more $A[i]=0$ indices

Solution choose k to minimize $(*)$

best choice: $\boxed{k = (\ln 2)\frac{n}{m}}$

$$\Pr[\text{false positive}] \approx \left(\frac{1}{2}\right)^k \approx (0.619)^{\frac{n}{m}}$$

This decreases exponentially in n