False Positive: We report $x \in X$, but it is not.

False Negative: We report $x \notin X$, but it is.

We will tolerate false positives (very few) but no false negatives.

Examples:
- $X = \{\text{weak passwords}\}$
  - Screen passwords for safety
  - Allow no weak passwords
  - May flag a few good passwords as being weak.

- $X = \{\text{URL's of malicious web sites}\}$
  - Allow no malicious sites
  - May flag a few safe ones.

Bloom Filter:
- 1970 by Burton Howard Bloom
- Can store very large sets $X$
- Answers queries in $O(1)$ time
- Uses $O(n)$ bits - $n = |X|$ (may be smaller than space needed to store all keys!)

Filtering:
- Given a large set $X$ of keys, answer membership queries is $x \in X$.

Objectives:
- Fast! $O(1)$ time
- Yes/No: No values/Just keys
- Errors allowed

Bloom Filter:
Let $X \subseteq U$ (universe) $|X| = n$ large!

Parameters: $k, m$ - TBD

Bit Vector: $B[0..m-1]$

$k$ Hash Functions: $h_1, ..., h_k$
$\forall i \in [0..m-1]$ $h_i : U \rightarrow [0, ..., m-1]$

$\text{Think: } h_i \text{ maps keys to random locations in bit vector }$
Controlling False Positives:

**Obs:**
- M - Bigger is better (but more storage)
- K - Trickier to balance

**Math Facts:**
- If |x| is small, $1 - e^{-x^2}$
- If an event occurs with probability $p$, the probability of $k$ independent occurrences is $p^k$.

**Analysis:**
- Assume $n = |X|, m = |B|, k = $ no. hashes
- Prob of hitting an arbitrary loc. of B, $Pr[j]$, with any hash is $1/m$.
- Prob of missing is $1 - 1/m$. So:
  $Pr(h_i(x) \neq j) = 1 - 1/m$
- After inserting all $n$ keys, each with $k$ hashes, prob of missing $B[j]$ is:
  $Pr(B[j] = 0) = (1 - 1/m)^k$
- Assuming $m$ large $\Rightarrow$ small $\Rightarrow$
  $Pr(B[j] = 0) \approx (e^{-1/m})^k$
- Assuming $m$ large $\Rightarrow$ small $\Rightarrow$
  $Pr(B[j] = 0) \approx e^{-kn/m}$

**Partial Correctness:**
- If $x \in X$, all hash locations $B[h_i(x)]$ set to 1 $\Rightarrow$ true
- If $x \notin X$, if any hash loc. $B[h_i(x)]$ 0 $\Rightarrow$ false
  $\Rightarrow$ But by coincidence, all may be set to 1 by other keys $\Rightarrow$ true (false positive)

**Bloom Filters II**

**Define:** $p = e^{1/m}$
- $kn/m$

(Later, we'll show best $p = \sqrt{2}$)
- Prob that any entry $B[j] = 0$ is $p$ $\Rightarrow$ Expected num. of 0's is $mp$
- Useful fact from prob. theory (Concentration about mean)
  - If $m$ large - actual num. of 0's is very nearly $mp$

**In summary:** $Pr[FP] = (\frac{1}{2})^{(\frac{1}{2})^{\ln(m)n}}$

To achieve a false pos. rate of $d > 0$
set $m = 1g \frac{\ln 1}{\ln 2}n$

**Equiv:** Num of bits per key is
$\frac{m}{n} \approx \frac{\ln \frac{1}{d}}{\ln 2} = O(\log \frac{1}{d})$

**False Positive Probability:**
- False positive (FP) occurs if all $h_i(x)$ set to 1 by other keys
- Since prob $B[j] = 0$ is $p$, we have
  $Pr[FP] = Pr[B[h_i(x)] = 1]^k$
  $= (1 - p)^k$
- To simplify $\Rightarrow$ Take $\ln$
  $\ln (Pr[FP]) = k \ln (1 - p)$
- By def. of $p$, we have:
  $\ln p = -kn/m \Leftrightarrow k = -\frac{m}{n} \ln p$
  $\Rightarrow \ln (Pr[FP]) = -\frac{m}{n} \ln p \cdot \ln (1 - p)$
- How to set $k$ to minimize this?
- Assume $m, n$ fixed, but $p$ varies
  $\Rightarrow \ln p \cdot \ln (1 - p) \approx \ln p \cdot \ln (1 - p)$
  $\Rightarrow \min_k Pr[FP] = (1 - p)^k \approx (\frac{1}{2})^{\frac{1}{2} \frac{\ln 2}{\ln 2}} = 0.618$