Bitmaps and Bloom Filters
Redis

http://redis.io

Binary Vectors in Redis

- Redis support bit level operations
- Commands SETBIT and GETBIT let you manipulate bit locations in a bit sequence, starting at 0
- Commands BITOP, BITCOUNT, and BITPOS operate on groups of bits
  - SETBIT subscribers 0 1
  - SETBIT subscribers 98 1
  - BITCOUNT subscribers
  - SETBIT visitors 98 1
  - GETBIT subscribers 3
  - BITOP AND sub:visitors subscribers visitors
  - BITPOS sub:visitors 1

Membership queries

- Improve searches with data structures that check for the nonexistence of an item in a set.
- Can return false positives but guarantee no false negatives.
- Approximate set membership problem
- Trade-off between the space and the false positive probability .
- Generalize the hashing ideas.

Approximate set membership problem

- Suppose we have a set
  $S = \{s_1, s_2, \ldots, s_n\} \subseteq$ universe U
- Represent S in such a way we can quickly answer "Is x an element of S ?"
- To take as little space as possible ,we allow false positive (i.e. $x \notin S$, but we answer yes)
- If $x \in S$, we must answer yes .

Bloom filters

- Originally developed by Burton Howard Bloom in 1970 for spell-checking applications
- Consist of an arrays A[n] of n bits (space) , and k independent random hash functions

  $h_1,\ldots,h_k : U \dashrightarrow \{0,1,..,n-1\}$

1. Initially set the array to 0
2. $\forall\ s \in S$, $A[h_i(s)] = 1$ for $1 \le i \le k$

   (an entry can be set to 1 multiple times, only the first times has an effect)
3. To check if $x \in S$, we check whether all location $A[h_i(x)]$ for $1 \le i \le k$ are set to 1

If not, clearly $x \notin S$.

If all $A[h_i(x)]$ are set to 1, we assume $x \in S$

Bloom filter example

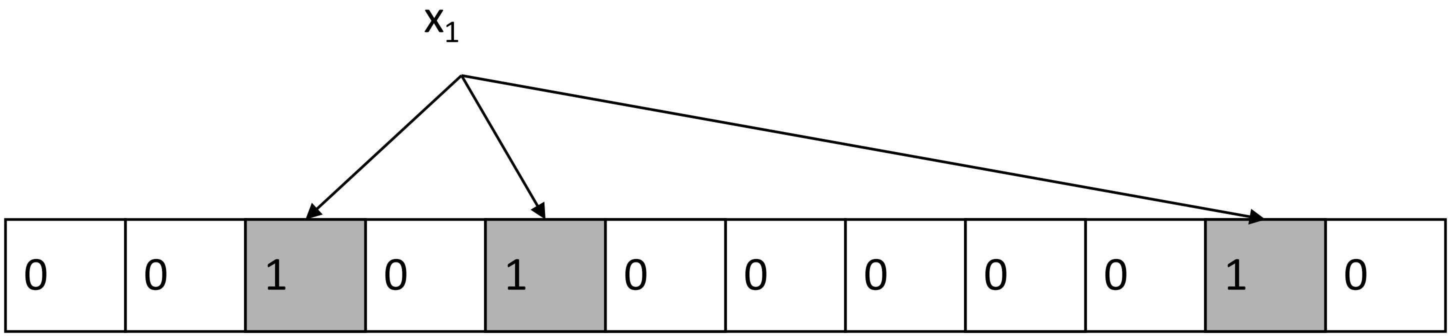
Consider k=3 independent hash functions
Bloom filter size n=12 bits
Possible number of elements m

0	0	0	0	0	0	0	0	0	0	0	0

Initially all positions are 0

Bloom filter example

Insert X_1

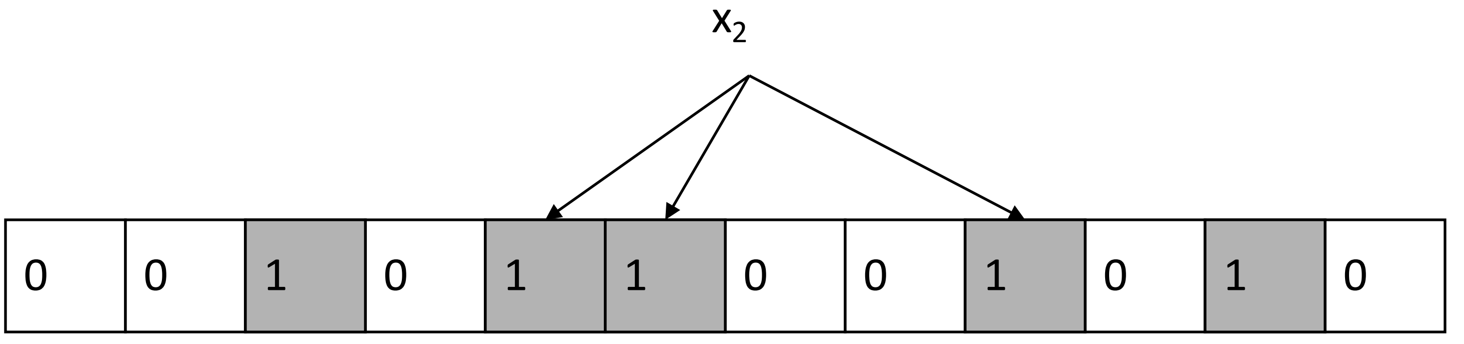


Each element of S is hashed k times

Each hash location set to 1

Bloom filter example

Insert X_2

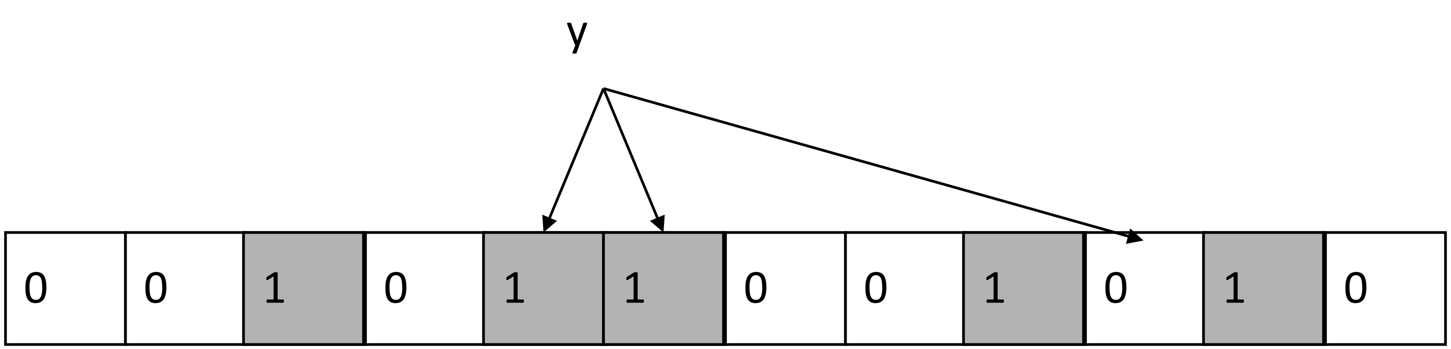


Each element of S is hashed k times

Each hash location set to 1

Bloom filter example

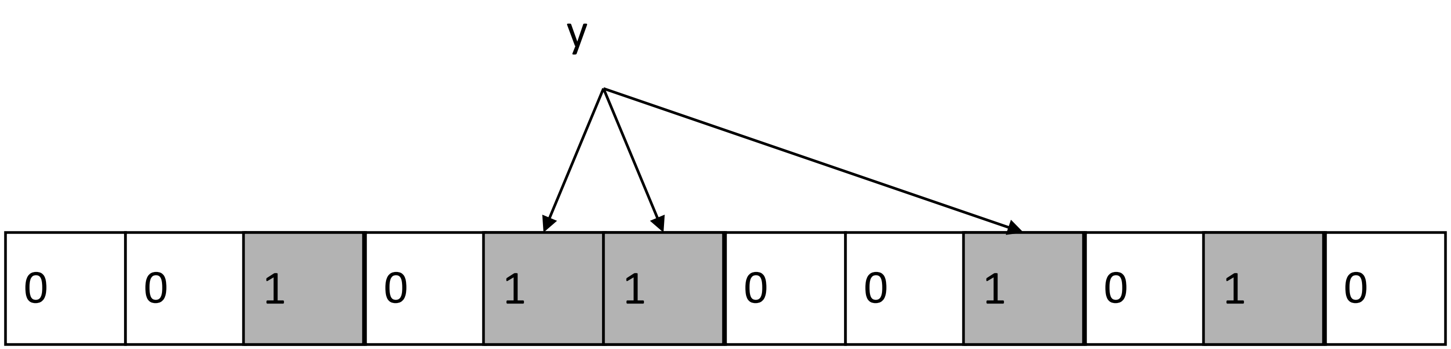
Query y



To check if y is in S, check the k hash location. If a 0 appears , y is not in S

Bloom filter example

Query y



If only 1s appear, conclude that y is in S
This may yield false positive

The probability of a false positive

- We assume the hash function are random.
- After all the elements of S are hashed into the bloom filters ,the probability that a specific bit is still 0 is

$$p = (1 - \frac{1}{n})^{km} \approx e^{-km/n}$$

To simplify the analysis ,we can assume a fraction p of the entries are still 0 after all the elements of S are hashed into bloom filters.

Probability of a false positive

- The probability of a false positive f is

$$f = (1-p)^k \approx (1-e^{-km/n})^k$$

- To find the optimal k to minimize f .

Minimize f iff minimize g=ln(f)

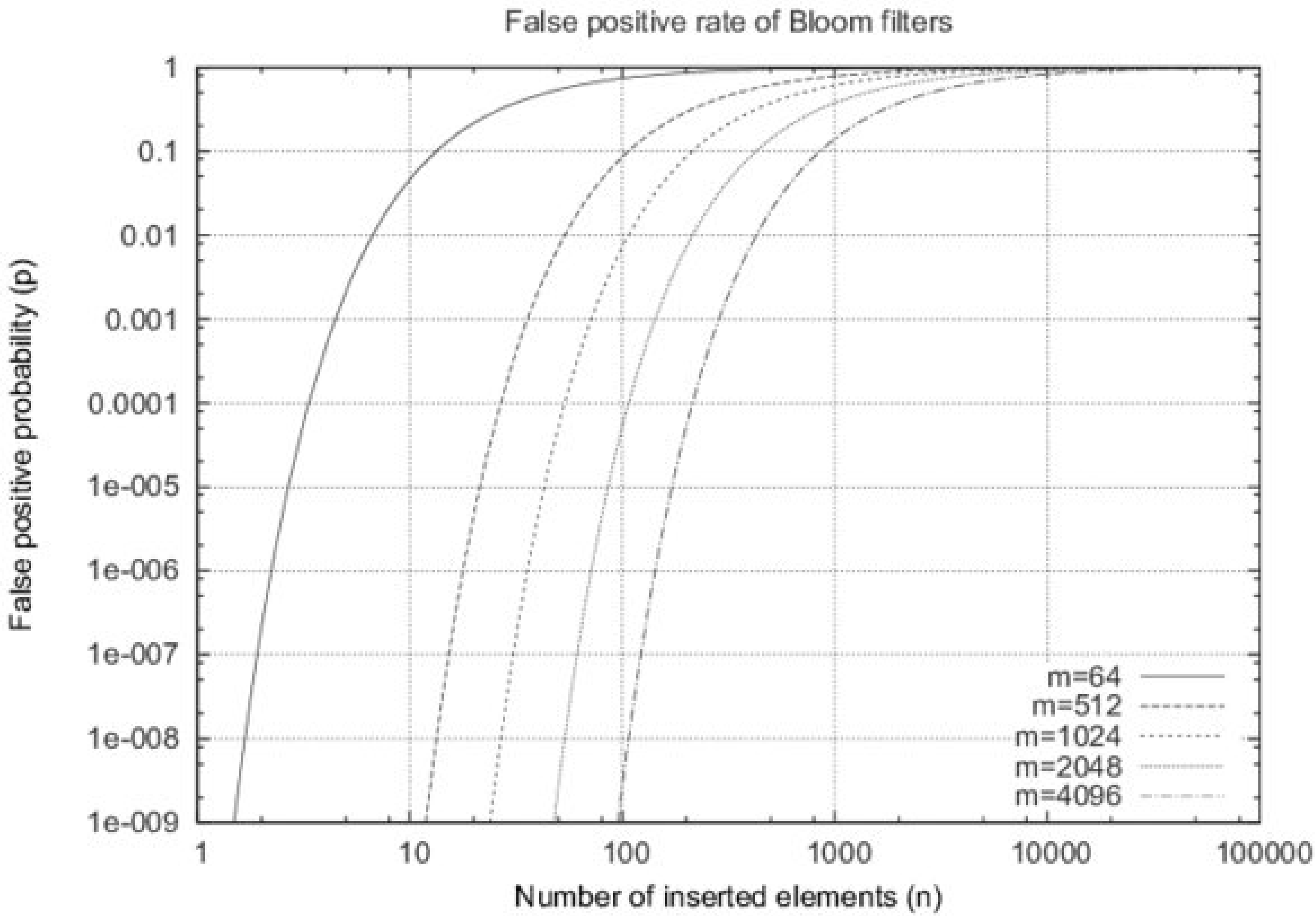
$$\frac{dg}{dk} = \ln(1-e^{-km/n}) + \frac{km}{n}\frac{e^{-km/n}}{1-e^{-km/n}}$$

$\Rightarrow$k=ln(2)*(n/m)

$\Rightarrow$f = $(1/2)^k$ = $(0.6185..)^{n/m}$

The false positive probability falls exponentially in n/m ,the number bits used per item !!

False positive rate for Bloom Filters



Bloom filter remarks

- A Bloom filters is like a hash table ,and simply uses one bit to keep track whether an item hashed to the location.
- If k=1 , it's equivalent to a hashing based fingerprint system.
- If n=cm for small constant c, such as c=8 ,then k=5 or 6 ,the false positive probability is just over 2% .
- It's interesting that when k is optimal

  $k=\ln(2)*(n/m)$, then p= 1/2.

  An optimized Bloom filters looks like a random bit-string

Redis modules

- Check out redis modules for third-party libraries and enhancement you can use in your projects
- https://redis.io/modules

For today…

- Finalize the groups
- Decide on the dataset you'll use for your program
- One person from your team should submit the ICON survey to describe the project:
  - Title of the project and short description
  - Dataset you plan to use – provide link and/or stats
  - Databases you plan to use (at least two)
  - Queries/analysis you will perform (at least two)