Preface to bloom filter

Q: Compare two sets

How can we create a function used to judge list A ?= list B ? List A and B are made up of unique elements.

1. Using sort comparison:
   Copy A and B to arrays/vectors

   Sort them

   compare one by one:

   Time complexity: $O(n \log n)$, Space complexity: $O(n)$
2. Hashing

   Create a hash set S for A

   For each element in B check whether they are in S or not

   Time complexity: $O(n)$, Space complexity: $O(n)$
3. methods with probability to be wrong:
   Compare sum $\sum_{i,i\in A} i \stackrel{?}{=} \sum_{i,i\in B}$,
   Compute product $\prod_{i,i\in A} i \stackrel{?}{=} \prod_{i,i\in B}$

We call the reduced sum / product as **"sketch"** of set A and B.

A wish to use the power of probability

Can we define a randomized `isEqual(A, B)` which satisfies:

$$\begin{cases} P\{isEqual(A,B)\} = 1 & if\ A = B \\ P\{isEqual(A,B)\} \to 0 & if\ A \neq B \end{cases}$$

A naive solution

Suppose we draw a hash function $h : u \to \{0, \ldots, 2^d - 1\}, h \in \mathcal{H}$, and $\mathcal{H}$ is a ROM. Define the following sketch for A and B:

$$S_A = \sum h(a)$$
$$S_B = \sum h(b)$$
$$isEqual(A,B) = S_A \stackrel{?}{=} S_B$$

If length of sets: $|A| \neq |B|$, then they are easily differentiable.

Otherwise:

1. If A is a permutation of B -> IsEqual(A, B) = true
2. If A is not a permutation of B:

suppose $a^* \in A \setminus B$ (set difference), then:

$$
\begin{aligned}
&P\{\sum h(a) = \sum h(b)\} \\
&= P\{\underbrace{h(a^*)}_{L} = \underbrace{\sum_{b\in B} h(b) - \sum_{a\in A\setminus a^*} h(a)}_{R}\} \\
&= 1/2^d
\end{aligned}
$$

because:

1. R is a random variable, independent of $h(a^*)$
2. L and R are independent, supported by the random oracle model
3. L is uniformly distributed

Summary

This question introduces us to bloom filters.