WK 40 • 275-090

Streaming Algorithms

OCTOBER 02 SATURDAY

Reservoir Sampling

- How to pick a random elements of an array?
Ans: choose a random integer and pick the element at that index.

- How to pick 'S' random elements from an infinite stream of elements?
  - Maintain a set of 'S' elements that may be updated every time a new element is added to the stream.

- At any given point, each element that has been seen so far should be equally likely to be in the set of s elements.

⇓ How?

- Save the first S elements

03 SUNDAY

- Suppose we have seen the first n-1 elements and now n^{th} element arrives.

- with probability S/n, keep n^{th} element and replace a randomly chosen saved element with the n^{th} element.

- otherwise discard the n^{th} element.

2021 Induction :- After n elements, the sample contains each element seen so far with probability S/n

OCTOBER

01 OCTOBER

FRIDAY

Problem statement:

Given a stream of size N which is too large to fit into memory, randomly select K elements such that each element has an equal probability of being chosen.

08 281-084 • WK 41

OCTOBER

FRIDAY

Bloom filter

Ex: Create gmail account → Enter user name → everytime (user name already exist)

⇓

all procedure done quickly

⇓ to check

{ linear search ?
{ Binary search ?

time consuming process

⇓ use

Bloom filter

- Bloom filter is a space efficient probabilistic data structure that is used to test whether an element is a member of a set.

for ex checking availability of username is a set membership problem, where the set is a list of all registered username

- The problem with the bloom filter is that they are probabilistic in nature that means, there might be some false positive results.

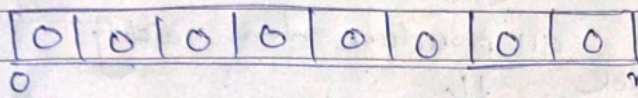
False Positive : It might tell that given username is already taken but actually it's not.

2021

THURSDAY

Working of Bloom filter

- An empty bloom filter is a bit array of m bits. all set to 'zero', like this



- we need 'k' numer of hash function to calculate the hashes for a given input.

- when we want to add an item in the filter, the bits at k indices $h_1(x), h_2(x), \ldots h_k(x)$ are set, where indices are calculated using hash functions.

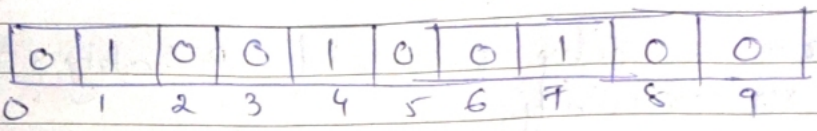
Example

- Suppose we want to enter 'throw' in the filter, we are using 3 hash functions & a bit array of length 10, all set to 0. initially.

- First we will calculate the hashes as following suppose,

h_1 ('throw') % 10 = 1
h_2 ('throw') % 10 = 4
h_3 ('throw') % 10 = 7

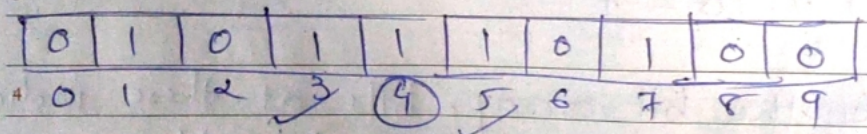
Now, we will set the bits at indices 1, 4, 7 to 1.



Again we want to enter 'catch', similarly we will calculate hashes

h_1('catch') % 10 = 3
h_2('catch') % 10 = 5
h_3('catch') % 10 = 4

set the bits at indices 3, 5 & 4 to 1.



Now, if we want to check 'throw' is present in filter or not, we will do the same process but in reverse order. Calculating respective hashes using h_1, h_2 & h_3 & check if all indices set to 1 in the bit array

- If all bits are set then we can say that 'throw' is 'probably present',
- If any of the bit at these indices are 0 then 'throw' is definitely not present.

TUESDAY

False Positive in Bloom Filters:

- The question is why we said 'Probably present' why this uncertainity.

* Let's take an example:

Suppose we want to check whether 'cat' is present or not. We will calculate hashes using h_1, h_2 & h_3.

check array.

h_1('cat') % 10 = 1 — set
h_2('cat') % 10 = 3 — set
h_3('cat') % 10 = 7 — set

If we check the bit array, bits at these indices are set to 1 but we know that 'cat' was never added to the filter. Bit at index 1, 3 & 7 was set when we added 'throw' & bit 3 was set when we added 'catch'.

Control probability of getting false positive ⟹ by controlling size of Bloom filter.

↓ probability of false positive ⟹ then ↑ no. of hash functions.

MONDAY

24 25 26 27 28 29 30

Probability of false positive

$$P = \left(1 - \left[1 - \frac{1}{m}\right]^{kn}\right)^{k}$$

where, m = size of bit array
n = no. of expected elements to be inserted
k = no. of hash functions.

Size of Bit array = $m = -\frac{n \log p}{(\log 2)^2}$ ← probability

Optimal no. of hash functions : $k = \frac{m}{n} \log 2$

Interesting properties of bloom filters

- Bloom filters never generate false negative result i.e. telling you that a username does not exist when it actually exists.

- Deleting elements from filter is not possible because, if we delete a single element by cleaning bits at indices generated by k hash functions it might cause deletion of few other elements.

TUESDAY

Counting Distinct Elements in a stream

⇒ Flajolet - Martin Algorithm (FM Algorithm)

- FM algorithm approximates the no. of unique objects in a stream or a database in one pass.

- If the stream contains n elements with m of them unique, this algorithm runs in O(n) time & needs O(log m) memory.

- The FM algorithm is an algorithm for approximating the no. of distinct elements in a stream with a single pass.

- Space consumption logarithmic in the maximal no. of possible distinct elements in the stream.

Example

Determine the distinct elements in the stream using FM algorithm.

Input stream of integers X = 1, 3, 2, 1, 2, 3, 4, 3, 1, 2, 3, 1

Hash function, $h(x) = 6x + 1 \cdot \bmod 5$

2021

MONDAY

Step-1 : Calculating Hash function [h(x)]

$h(x) = 6x + 1 \bmod 5$

$h(1) = 6(1) + 1 \bmod 5$

$= 7 \bmod 5$

$\boxed{h(1) = 2}$

similarly, calculating hash function for the remaining input stream.

∴ $h(1) = 2$ $\quad h(4) = 0$

$h(3) = 4$ $\quad h(3) = 4$

$h(2) = 3$ $\quad h(1) = 2$

$h(1) = 2$ $\quad h(2) = 3$

$h(2) = 3$ $\quad h(3) = 3$

$h(3) = 4$ $\quad h(1) = 2$

Step-2 : Write Binary Equivalent for hash function calculated

$h(1) = 2 = 010$ $\quad h(4) = 0 = 000$

$h(3) = 4 = 100$ $\quad h(3) = 4 = 100$

$h(2) = 3 = 011$ $\quad h(1) = 2 = 010$

$h(1) = 2 = 010$ $\quad h(2) = 3 = 011$

$h(2) = 3 = 011$ $\quad h(3) = 3 = 011$

$h(3) = 4 = 100$ $\quad h(1) = 2 = 010$

Step-3 : Finding Trailing zeros

Now, write the count of trailing zeros in each hash function bit

2021

OCTOBER 09
SATURDAY

Not trailing zeros - if bit change then trailing zero consider

$h(1) = 2 = 010 = 1$	$h(4) = 0 = 000 = 0$
$h(3) = 4 = 100 = 2$	$h(3) = 4 = 100 = 2$
$h(2) = 3 = 011 = 0$	$h(1) = 2 = 010 = 1$
$h(1) = 2 = 010 = 1$	$h(2) = 3 = 011 = 0$
$h(2) = 3 = 011 = 0$	$h(3) = 4 = 100 = 2$
$h(3) = 4 = 100 = 2$	$h(1) = 2 = 010 = 1$

Step-4

Write the value of maximum no. of trailing zeros.

- The value of $r = 2$

- The distinct value $R = 2^r$

$= 2^2$

$= 4$

Hence, there are 4 distinct elements 1, 3, 2, 4

10
SUNDAY

PDF Created Using

Camera Scanner

Easily Scan documents & Generate PDF

https://play.google.com/store/apps/details?id=photo.pdf.maker