19AIE203 “Data Structures and Algorithms 2 ”

Project Report

Bloom Filters

Bachelor of Technology
in
Artificial Intelligence & Engineering

Submitted by:
Gandham Sai Ram Pavan - AM.EN.U4AIE20125

Subject Teacher
Mr.Sethuraman N Rao

AMRITA
VISHWA VIDYAPEETHAM

School of Computer Science & Engineering
Kollam, Kerala, India – 690 525.

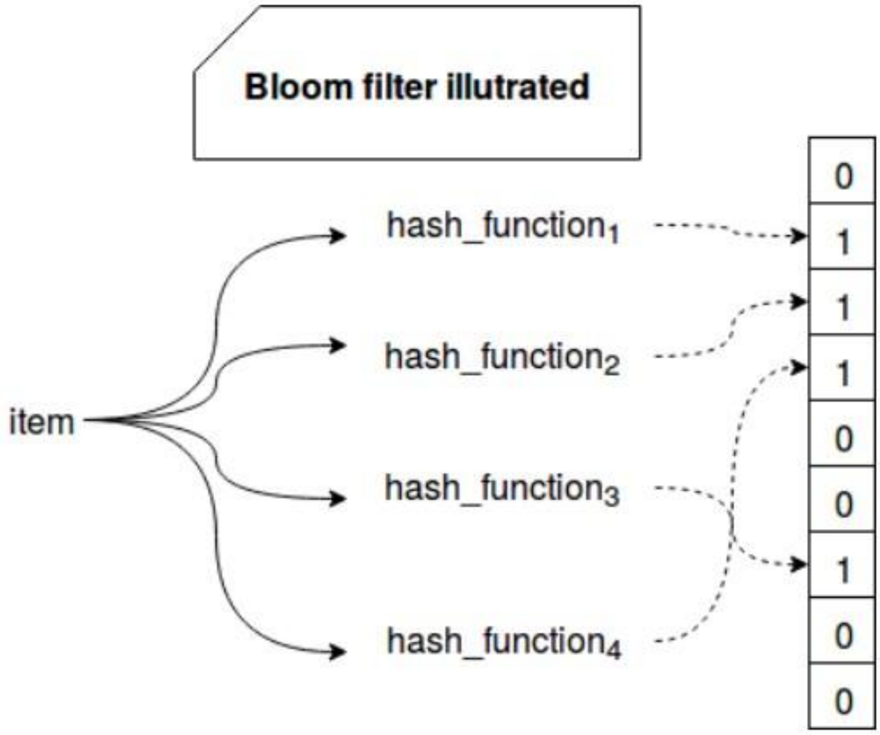
3rd Semester 2021

A Bloom filter is a space-efficient probabilistic data structure that is used to test whether an element is a member of a set. For example, checking availability of username is a set membership problem, where the set is the list of all registered usernames. The price we pay for efficiency is that it is probabilistic in nature that means, there might be some False Positive results. False positive means, it might tell that a given username is already taken but actually it's not.

IMPLEMENTATION OF BLOOM FILTER IN JAVA

The algorithm is based on a bit vector of size 6400 and 4 independent and uniformly distributed hash functions. When a new element is handled by the filter, it's hashed against each of the functions. Their results correspond to the bit vector indexes that will be set to 1. The same operation is made to check membership. The algorithm reads the values from the computed indexes and depending on the result, it tells whether:

- the element is certainly not in the set - if one of received values is equal to 0
- the element is probably in the set - if all received values are equal to 1



In our project, we used a bloom filter to make a sign-up interface. We have used several methods to implement this. Firstly, we created a bit array of length 6400 bits.
We then used four hash functions to hash each mail id into the bit array.

1.Hashing

We used a class hash function which has methods like calculateHash(), getHashValue(), and HashFunction() which uses the MAD method.

2.insert-val

This method takes an input which will be an email id, hashes it, adds it to the bit array and stores it in an external file .

3.check_value

This method checks the same process as insert_val and checks if the hash values are already marked as 1. If the values are the same, then it gives an output of "Email already Taken" or else, it shows "Email is available".

4.EmailValidator

This method checks whether the given input is valid mail or not.

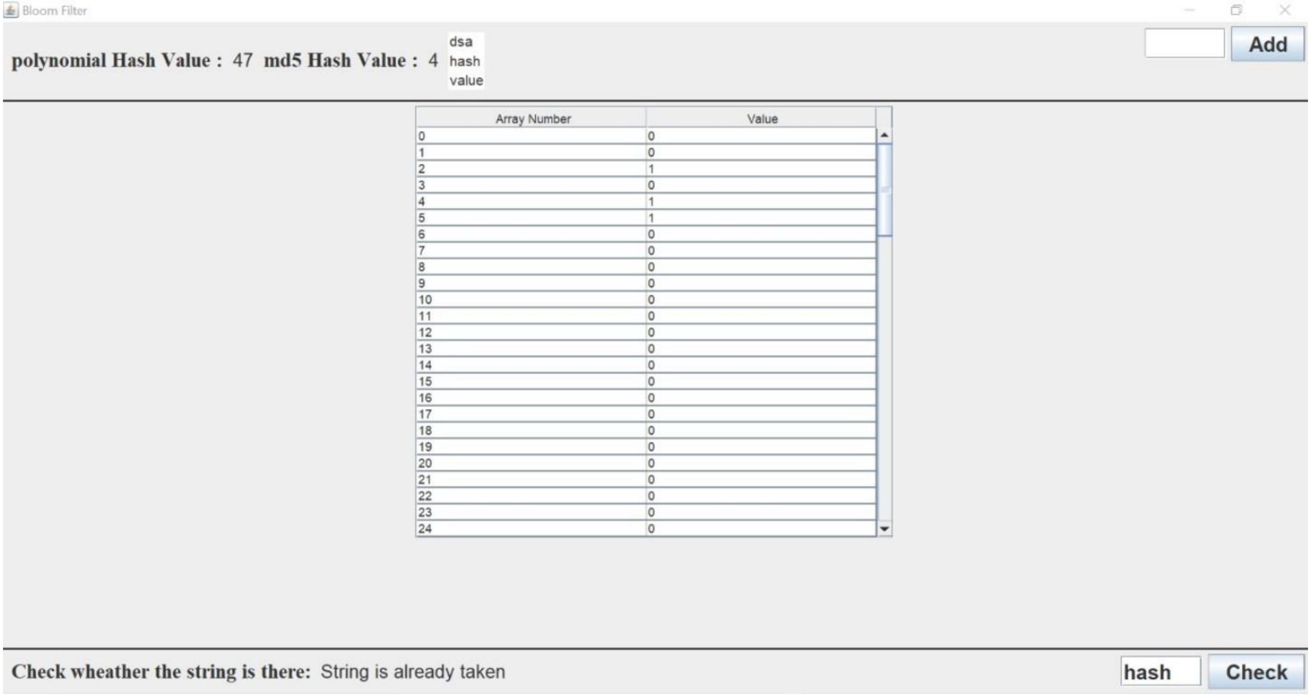
5.Size() and check_size()

These methods check the no.of inputs in the file and give out the size of the file and also checks if it is empty or not empty.

Then we used a switch statement to select an individual option.
1.insert
2.contains
3.chech empty
4.size

VISUALISATION OF BLOOM FILTER

The visualisation is done by SWING which is a GUI widget toolkit for Java. Mainly the code consists of three classes and two sub classes which is described below.



polynomial Hash Value : 47 md5 Hash Value : 4

dsa
hash
value

Array Number	Value
0	0
1	0
2	1
3	0
4	1
5	1
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0

Check wheather the string is there: String is already taken

hash

Constructor:

First we will initialize the array. We will make the frame and set its size. Here we will initialize all the labels,buttons,List and tables which are required for the project with their specific use. Depending on the current use case we could click the buttons to either add the string to the array or to check whether the string is already taken or not. The string which we are adding to the array could be seen as they are parallelly added to the list also. The table has two columns; they are Array numbers and Corresponding values. Initially all the values in the table will be zero as no string will be there in the array but if any string is added then corresponding array number values will be changed to 1. Every value in the table could be seen by just scrolling the table.

Hash Function :

If we want to hash any object we should implement two code portions of code. First one is the Hash code and the second one is the compression function.

Hash Code- The first action that a hash function performs is to take an arbitrary key k in our map and compute an integer that is called the hash code that need not to be in the range of Non zero positive elements and it could be any integer.

Compression Function- Once we have determined an integer hash code for a key object k, then we will map that integer to the array that is the compression function.

Here in our program we are using two hash code:

1)Polynomial Hash
2)MD-5
We will use common compression function that is MAD(Multiplication Add Division)

Insert Function:
If the user wants to insert any particular string. First we will calculate the hash value of that particular string. The hash value which was computed for the string at that current hash position in the array we will make it as 1. This will be depending on the number of hash functions. In our code we are using two hash functions so for one word to be inserted into the array we have to make two positions in the array as 1.

Check Function:
To check whether the string is taken by the user. First we will calculate the hash value of that particular string. We will check at that hash value whether the array is having 1 or not if it is having 1 then the string has been taken by some one. If the array is having 0 at that current hash value position then we could tell that no one has taken that particular string.

Time and Space Complexity

- The query response of Bloom Filter is very fast, and it is in O(1) time complexity.
- The space complexity of the Bloom Filter having n elements has O(n).
- The most significant advantage of Bloom filters over other data structures such as self-balancing trees, tries, HashMaps is in terms of space utilization.
- Time Complexity comparison

1. Tries: O(string length)
2. BST: O(string_length * height)
3. Bloom: O(1) or O(hash function)