CS515: Computer System Lab 2

Date: 3rd Mar 2022 Assignment 6
Submission Filename: assign6.c or assign6.cpp and assign6README.txt Due Date: 7th Mar 2022 9 am

1 Overview

The learning objective of this assignment is for students to gain some programming experience on designing a *Bloom Filter* which is commonly used to eliminate unnecessary accesses to slower storage or expensive lookups. *Bloom Filter* is a data structure for storing an *unordered* set using *hash* functions.

Suppose we are given a set S, drawn from a universe U and queries of the form `is` $x \in S$`?`, we **define** a Bloom filter as follows:

- Take an array A with n elements and take m hash functions h_1, h_2, $\cdots$, and h_m. Each hash function maps the universe U to $[0, n-1]$.
- Suppose we start with S being the emptyset, then $A[i]$ is set to 0 for all i.
- Now, suppose we insert an element e into the set S. We set $A[h_1(e)]$, $A[h_2(e)]$, $\cdots$ $A[h_m(e)]$ all to 1
- If any of these m positions were already set to 1 then we just leave it at 1.

Membership Queries: Given a query `is` $x \in S$`?` we look at $A[h_1(x)]$, $A[h_2(x)]$, $\cdots$, $A[h_m(x)]$. If they are all 1 then we return `yes` as the answer, otherwise we return `no`.

Note that it is possible that $x \notin S$ but $A[h_1(x)]$, $A[h_2(x)]$, $\cdots$, $A[h_m(x)]$ are all set to 1 (due to some other elements being in S which have set those positions to 1). This phenomena is called a `false positive` i.e. the query returns a `yes` answer although the element is not in the set.

2 Task to be carried out

For this task you will implement a Bloom filter on the universe $U = \{0, 1, 2, ..., n-1\}$. Your array should have size n. You will choose any m hash functions of the from $h(x) = ((ax + b) \mod c) \mod n$ where a, b and c are large positive integers.

Now, you should do the followings:

1. **Initialization of Bloom Filter**: Initialize your Bloom Filter by inserting any k elements at random from U. Let us denote this randomly generated k elements as S. Display the Bloom filter created using S.
2. **Membership Queries & Finding Causes for False Positives**: Once the state of the Bloom filter has been displayed, then the program should ask to move forward with a keypress. After that the program should launch queries on each of the remaining $n - k$ items. If any one of these returns a `yes` answer, the program should pause and find out why this *false positive* has been returned i.e. it should locate the hash function collisions that have caused this *false positive* to occur. More formally, if we get a *yes* answer for x although x is not in S, this means that $A[h_1(x)]$, $A[h_2(x)]$ and $A[h_3(x)]$ are all 1. Your program should return (at most) m elements u_1, u_2, $\cdots$, u_m from S which hash to the positions $h_1(x)$, $h_2(x)$, $\cdots$, $h_m(x)$ through one of the m hash functions. Your program should display which location has been set to 1 by which element of S through which hash function.
3. **Summarizing False Positive Cases**: Finally the program should count the total number of `false positives` occurring in all $n - k$ membership queries and display the `false positive` probability p i.e. the fraction of `false positives` out of $n - k$. Formally p can be defined as $p = \forall x \in \{U - S\}(\frac{count((x \in S) = yes)}{|U| - |S|})$
4. **Input File Format**: The entries of the input file is as follows-

```
m //no of hash functions
a_1 b_1 c_1 // a b c parameters of hash function 1
a_2 b_2 c_2 // a b c parameters of hash function 2
...
a_m b_m c_m // a b c parameters of hash function m
n // The array size of the bloom filter
k // No. of elements to be inserted
S // elements of S
```

5. **Output File Format**: The entries of the output file will be as follows-

```
m =
n =
k =

The Bloom filter is
//Should generate the sequence of n length binary string

Membership queries x ∈ S  ∀x ∈ {U − S}
x, false positive (y/n), collusion cases

Fraction of false positives, p =
```

3 Assumption

We will assume that there is no deletion for the purposes of this assignment i.e. only insertion and membership queries have to be implemented

4 Submission

You need to implement the bloom filter either using `c` or `c++`. Your need to include the followings in your submission file-

- Your C or C++ code
- A README file (*assign6README.txt*) comprising the necessary documentation required to execute your program must also be created. You must also mention the input filename and output filename. Try to generate the output filename using the combination of m, n and k values. For example, if $m = 3$, $n = 100$ and $k = 25$ then the output filename could be *op_m3n100k25.txt*.

5 Guidelines

Copying programs from others or from other sources and allowing others to copy your program will be equally penalized.