For Lab 5, you will implement a Bloom Filter and study its performance on the task of storing IP addresses. You will compare its runtime with the provided chain_hash program which implements hashing with chaining.

1 Implementing a Bloom Filter

The provided files ip_address_10k.txt and ip_address_100k.txt contain 10000 and 100000 distinct IP addresses respectively. Write a program that implements a Bloom Filter with \( k \) hash functions. Use the implemented Bloom Filter to maintain \( N \) IP addresses, read from previous files. Each hash function should have the following form: hash value = \( \sum_{i=1}^{d} a_i \cdot IP_i \mod M \), where \( IP_i \) is the \( i^{th} \) component of an IP address, and \( a_i \) is an integer selected uniformly at random (just once, not every time) between 0 and \( M - 1 \). Then, estimate the false positive rate of the Bloom Filter by checking the IP addresses from the test sets: ip_addr_test_1k.txt and ip_addr_test_10k.txt (note that the test sets and inputs have no IP addresses in common which you can check using the chain_hash program). You must also measure the time required to insert and check membership of the Bloom Filter.

2 Analysis and Comparison

- Run your program 5 times for each setting of \( M, N \) and \( k \), where \( M = 60013 \), \( N = 1000, 2000, 4000, 8000, 10000 \), and \( k = 1, 3, 4, 6 \) with ip_address_10k.txt as input and ip_addr_test_1k.txt as test.

- Run your program for the following settings of \( M, N \) and \( k \): \( M = 600043 \), \( N = 50000, 70000, 100000 \), and \( k = 1, 3, 4, 6 \) with ip_address_100k.txt as input and ip_addr_test_10k.txt as test.

- Record the time used to insert all elements in the hash table and the average false positive rate of each setting.

- Then, compare the insertion time and cache check time of the Bloom Filter with the method of hashing with chaining. For example, you can execute the chain_hash program using the following commands

  ```
  ./chain_hash ip_address_10k.txt ip_addr_test_1k.txt 60013 N 1000 0,
  ```

  for all \( N = 1000, 2000, 4000, 8000, 10000 \).
Finally, create a table of all your results and use your favourite plotting software to create three plots:

- Insertion time of Bloom filter with $k = 3$ and hashing with chaining, for all the sizes listed.
- Cache check time for the same settings.
- False positive rates in each setting.