# Bit String Array

Khudayar Farmanli
*Politecnico Di Torino*
*Data Science and Engineering*
*Computer Aided Simulations and Performance Evaluation*
Torino, Italy
s276310@studenti.polito.it

***Abstract*—The objective of this homework is the application of bit string hashing and calculating theoritical probabilities, memories taken and observing its differences and similarities with respect to fingerprinting method.**

## I. Input Parameters

- Dataset - contains various characteristics related different movies, we just take the rows contain ”IT” as region then eliminate all the other things and take the ”title” part.
- m - the number of elements in final set which is 176339 in our particular case, there were more but when I create set it directly eliminated the duplicates, probably more pre-processing is needed but for our purpose it is not crucial.
- b - predefined number of bits for further processing, in our case [19, 20, 21, 22, 23, 24, 25, 26]

## II. Output parameters

- Probability of False Positive for Bit String Arrays - calculated based on below formula:

$$Pr(FP) = \frac{NumberOf"1"bits}{2^b}$$

- Probability of False Positive for Fingerprint sets - calculated based on below formula:

$$Pr(FP) = 1 - (1 - \frac{1}{n})^m$$

where $n = 2^b$.

- Memory - taken by bit string arrays and fingerprint sets.

## III. Main data structures

For developing this simulation I have used python sets, lists and numpy arrays. I have stored Fingerprints to sets, Bit string arrays to lists and generated group of bits with the help of numpy arrays.

## IV. Results

The aim of this study was to calculate False Positive probability based on different number of bits which had been previously defined for both Bit String Arrays and Fingerprint sets and see the similarities and differences. Also, the memories taken by both method calculated and visualized for comparison. The results I obtained are presented below:

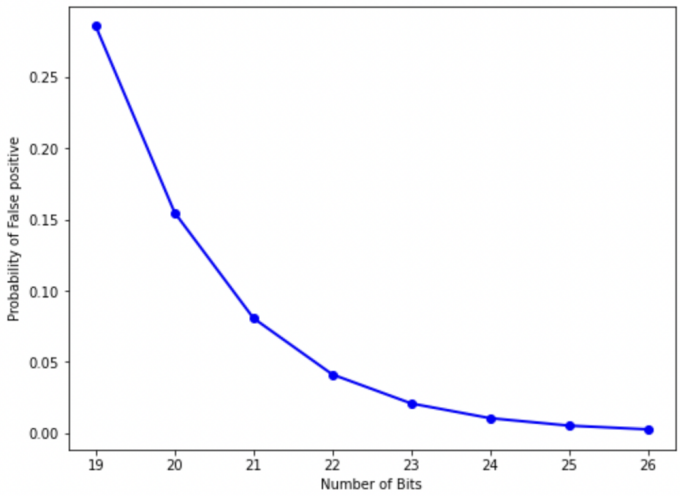


Fig. 1: The graph shows how different Bit String Arrays result different False Positive probability based on particular bits.

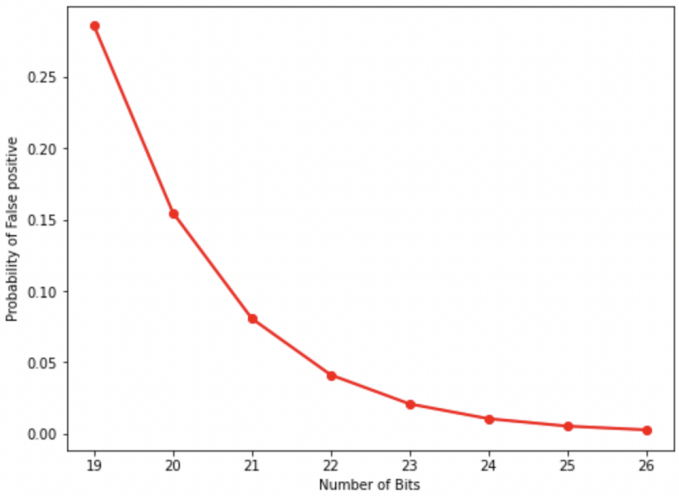


Fig. 2: The graph shows how different Fingerprint sets result different False Positive probability based on particular bits.

As it can be easily seen from the both above graphs, the increase in number of bits decrease the probability of false positive and the results provided by both method are very similar.

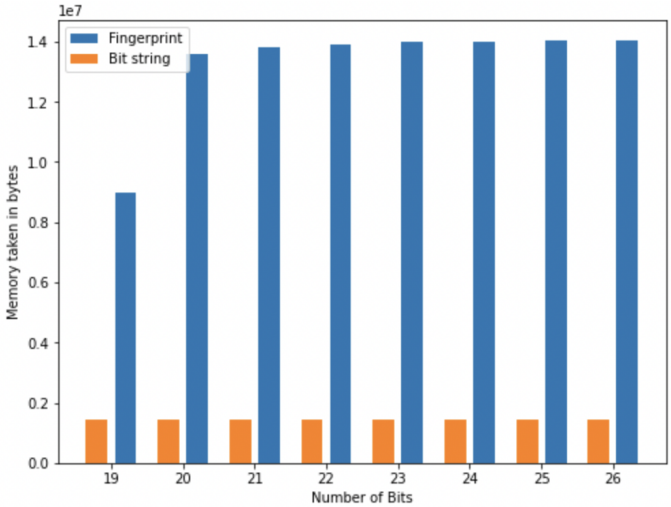


Fig. 3: The memory taken by Fingerprint sets and Bit string arrays based on particular number of bits.

From above graph we see that memory taken by Fingerprint set is lower for lower number of bits and higher for higher number of bits. Of course, it is understandable, because it is set and contains unique elements and in case of higher number of bits, it means low probability of false positive while encoding and getting more unique values. Also, it takes undoubtedly much memory for storage with respect to Bit String Arrays which only take small amount of memory.

## V. Conclusion

At the end, I obtained below conclusions based on my experiments:

- The probability of false positives based on particular number of bits are similar for both Bit string array and Fingerprint since I could not visualize them together, because it was not possible to see the difference with human sight. They were overlapping, only when checking numbers we see the difference which is in decimals.
- Memory used by Fingerprints are fairly higher than Bit string arrays, so the method introduced is really better in terms of memory usage.