Question1 :

To compute the expected number of set bits in b1 OR b2, consider that each bit in b1 OR b2 has two possible states: either it is set, or it is not set. The probability that a bit in b1 OR b2 is set depends on whether the corresponding bits in b1 and b2 are both set, one is set, or neither is set.

Case 1: Both bits in b1 and b2 are set

The probability that both bits in b1 and b2 are set is the product of the probabilities that each bit is set individually. Since the bits are set randomly with uniform probability, this probability is $(n/b)^2$.

Case 2: One bit in b1 and b2 is set

There are two ways in which one bit in b1 and b2 is set: either the bit in b1 is set and the bit in b2 is not, or the bit in b1 is not set and the bit in b2 is set. The probability of each of these cases is $(n/b)(1 - n/b)$.

Case 3: Neither bit in b1 and b2 is set

The probability that neither bit in b1 and b2 is set is the complement of the probability that both bits are set or one bit is set. This probability is $1 - (n/b)^2 - 2(n/b)(1 - n/b)$.

Expected Number of Set Bits

The expected number of set bits in b1 OR b2 is the sum of the products of the number of set bits in each case and the probability of that case. This can be expressed as:

$E(b1 \text{ OR } b2) = 2(n/b)^2 + 2(n/b)(1 - n/b)$

Substituting $b = 2^r$, we get:

$E(b1 \text{ OR } b2) = 2(n/2^r)^2 + 2(n/2^r)(1 - n/2^r)$

Sketch of the Problem Setting

Consider a row of b = 2^r bits, with n bits set: n bits in the first half b1 and n bits in the second half b2. The problem is to determine the expected number of set bits in b1 OR b2.

Conclusion

The expected number of set bits in b1 OR b2 is $2(n/2^r)^2 + 2(n/2^r)(1 - n/2^r)$.

$$E(b1 \text{ OR } b2) = 2(n/2^r)^2 + 2(n/2^r)(1 - n/2^r)$$

In this case, b = 128 × 512 = 65536, n = 30000, and r = 5.

$$E(b1 \text{ OR } b2) = 2(30000/65536)^2 + 2(30000/65536)(1 - 30000/65536) = 33368.84$$

Therefore, we expect approximately 33368.84 bits to be set in the rank i + 1 row for the given row.

Solution for Handling 1 billion Unique String Identifiers:

1. Generating and Partitioning Data:

Generate 1 billion unique string identifiers for books and partition them among the three branches: Central Campus, Sachsendorf, and Senftenberg. Save these identifiers into separate files for each branch.

2. Creating Bloom Filters:

Load the identifiers from the files corresponding to each branch and create Bloom filters for each branch using a scalable Bloom filter implementation.

3. Combining Bloom Filters:

Combine the Bloom filters from all branches into a centralized Bloom filter by taking the union of the individual Bloom filters. This centralized Bloom filter represents the combined set of book IDs across all branches.

4. Querying:

When a book ID is queried, check the centralized Bloom filter. If the query returns a positive match, it indicates that the book might exist in one or more branches. If it returns a negative match, the book definitely doesn't exist in the library.

Sketches:

Visual representations or diagrams illustrating the Bloom filters for each branch and the final combined Bloom filter would help visualize the distribution of book IDs across branches and the centralized lookup structure.