CS386D Database Systems
HW4 Solutions

1.

a)

Given that each value appears every m records, there will be a run of m−1 0's followed by a 1 repeated many times. It requires $\lceil 2 \log(m_i - 1) \rceil$ bits to represent such a run in bitmap. Each value will appear $\approx 10^8/m_i$ times.

Except primary key column, the number of bits = $2m_i$ * (# of tuples each value appears in T) * $\lceil \log(m_i - 1) \rceil$

So, the total number of bytes over 100 columns:

$[\sum_{i=1 \text{ to } 100} 2 * 10^8 * \lceil \log(m_i - 1) \rceil] / 8$ bytes = $25 * 10^6 \sum_{i=1 \text{ to } 100} \lceil \log(m_i - 1) \rceil$ bytes

b)

Since C_0 is a primary key and thus a unique value, every bit map will be of the form 0*1. Further, the set of bitmaps will be 0^i1, where 0^i means i zeros in a row, and there will be a bitmap for each i, i = 0 to the number of records - 1.

c)

$2m_i$ * (# of tuples each value appears in T) * $\lceil \log(m_i - 1) \rceil$

For table S, total number of tuples = n = 10^6

- For first 50 columns: m_i = n/1000 = 10^3, for next 50 columns = 10^4

Total number of bytes = $[\sum_{i=1 \text{ to } 50} 25 * 10^4 * \lceil \log(m_i - 1) \rceil] + [\sum_{j=51 \text{ to } 100} 25 * 10^4 * \lceil \log(m_j - 1) \rceil]$

= $25 * 10^4 [\sum_{i=1 \text{ to } 50} \lceil 9.96 \rceil + \sum_{i=51 \text{ to } 100} \lceil 13.29 \rceil]$

= $25 * 10^4 (10 * 50 + 14 * 50) = 25 * 10^4 * 1200$

= $3000 * 10^7$ = 300 MB

For table T, total number of tuples = n = 10^8

- For first 50 columns: m_i = n/1000 = 10^5, for next 50 columns = 10^4

Total number of bytes = $[\sum_{i=1 \text{ to } 50} 25 * 10^6 * \lceil \log(m_i - 1) \rceil] + [\sum_{j=51 \text{ to } 100} 25 * 10^6 * \lceil \log(m_j - 1) \rceil]$

= $25 * 10^6 [\sum_{i=1 \text{ to } 50} \lceil 16.61 \rceil + \sum_{i=51 \text{ to } 100} \lceil 13.29 \rceil]$

= $25 * 10^6 (17 * 50 + 14 * 50)$

= $3875 * 10^7$ = 38.75 GB

d)
Size of a data page = 4kB
Size of one record = Size of c_0 + size of m_i * 50 + size of m_j * 50 = 4 + 1250 + 1000 = 2254 bytes

For table S, n = 10^6
Storage require for table S = 10^6 * 2254 bytes = 2254MB
= 10^6 * 2254 / (4 * 1000) pages = 563,500 pages

For table T, n = 10^8
Storage require for table S = 10^8 * 2254 bytes = 225.4GB
= 10^8 * 2254 / (4 * 1000) pages = 56,350,000 pages

For this example, compressed bitmap representation is smaller than horizontal representation. The horizontal representation grows linearly in the size of the records. Although just two data points is not enough to make a general comment about growth rate, it is clear the benefit of compressed bitmap storage for the larger table, T, is much greater than for the smaller table S.

2.
a)
Keys are used to uniquely identify corresponding values. If they are stored separately, some pointer pointing to the value location must be used, which may lead to extra disk I/Os.

b)
i. Assumption: Each block has one Bloom filter. A block is 64 MB and a key, value pair requires 1024 bytes.
This leads to 64MB/1KB = 64K keys.
So, m = 64K

64MB of storage requires just 64K * 10 bits
Size of an effective filter for a block of 64MB storage = 2^{16} * 10 bits = 81,920 bytes

ii. Each server has: [Size of database / (Number of servers * Size of a block)] blocks
= $8*2^{40}$ / ($128* 64 * 2^{20}$) = 1024 blocks

Number of bytes per server with 1024 blocks = 1024 * 81920 = 83,886,080 bytes

iii. Number of hash functions k = ln (2) * m / n = ln 2 * (10 * 2^16 / 2^16) = ln (2) * 10 ≈ 7

iv. Probability of a false positive = $(½)^k$ = $(½)^7$ ≈ 0.78%