Name: Xiang Gu
EID: xg2847
HW4

1.
(a). Assuming each value of a particular column is assigned to records in a round-robin fashion, we know from the textbook exercise that the number of bytes needed for all the bitmap indexes for column c_i is 2 * n * ceil(log_2(m_i - 1)) / 8. Thus, the number of bytes needed for all columns is **25,000,000 * sum_{i=1}^100 ceil(log_2(m_i - 1))**.

(b). **No.**

(c).
(i). For table S, plug in the numbers we get 2 * 1,000,000 / 8 * (50 * ceil(log_2(1,000 - 1)) + 50 * ceil(log_2(10,000 - 1))) = 250,000 * 50 * (10 + 14) \appro **300MB**
(ii). For table T, plug in the numbers we get 25,000,000 * 50 * (17 + 14) \appro **38.75GB**

(d).
(i). For table S, 1,000,000 * (4 + 50 * 25 + 50 * 20) \appro **2.254GB**
(ii). For table T, 100,000,000 * (4 + 50 * 25 + 50 * 20) \appro **225.4GB**
(iii). For table S, 2.254GB/4KB = **578 pages**
(iv). For table T, 225.4GB/4KB = **57800 pages**

2.
(a). It's because keys are the unique identifier to know which record the value belongs to.

(b).
(i). # of key-value pairs in a block = 64MB/1KB = 64 * 2^10
Hence # of bits per filter = 10 * 64 * 2^10 = **640K bits**

(ii). The memory per server = 8TB/128 = 64GB
hence # of blocks in a server = 64GB/64MB = 2^10
So # of bits per server to represent the filters = 640K * 2^10 = **640M bits**

(iii). k* = ln2 * m/n = ln2 * 10 = 6.931
Hence the optimal number of hash functions should be **7**.

(iv). Pr(false positive) = (1 - (1 - 1/m)^(kn))^k \appro (1 - e^(-kn/m))^k = (1 - e^(-7/10))^7 = **0.008193**