CS4224/CS5424 Quiz 3: Storage & Indexing **Wk 4, Sem 1, 2023/24**

For questions 1 & 2 on LSM storage, assume that timestamp($S_{i,j}$) < timestamp($S_{i,k}$) if $j < k$.

1. Consider the following LSM storage for a relation that is compacted using STCS with a threshold of 3 (i.e., a compaction is triggered at a tier if it contains 3 SSTables). Show the state of the LSM storage after a compaction at tier 1.

Tier 1 SSTables

$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
2, m	20, a	9, u
8, n	38, ⊥	21, v
12, ⊥	94, b	37, w
38, o	102, c	38, x
87, p	133, d	102, y
150, q	220, e	110, ⊥

Tier 2 SSTables

$S_{2,1}$
4, q
8, u
12, r
37, s
39, z
40, t
67, c
110, u
148, x
170, v
200, a
211, b
237, i

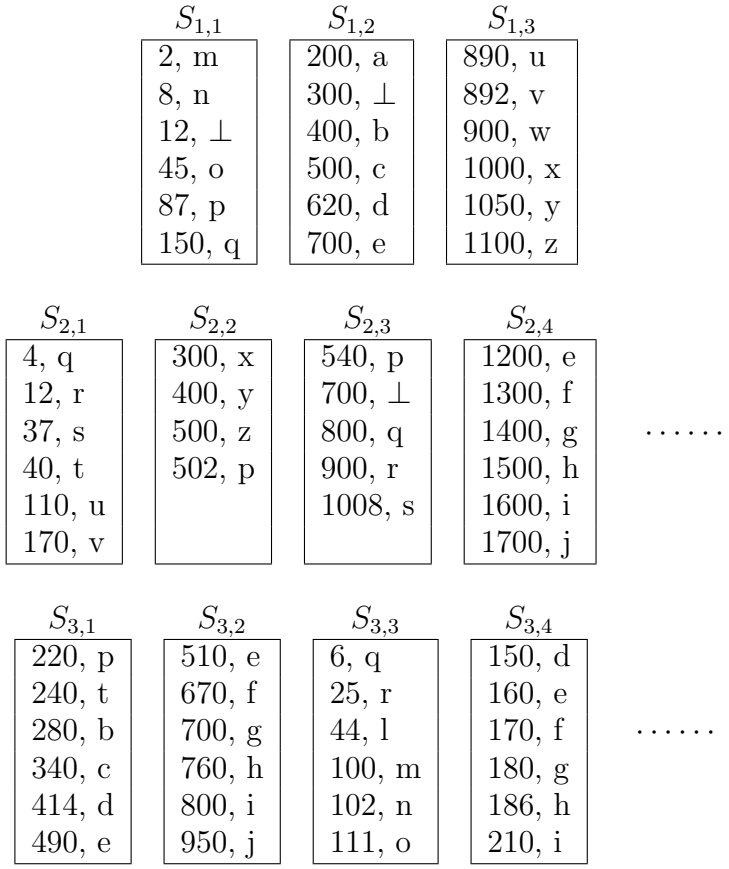
Solution:

Tier 2 SSTables

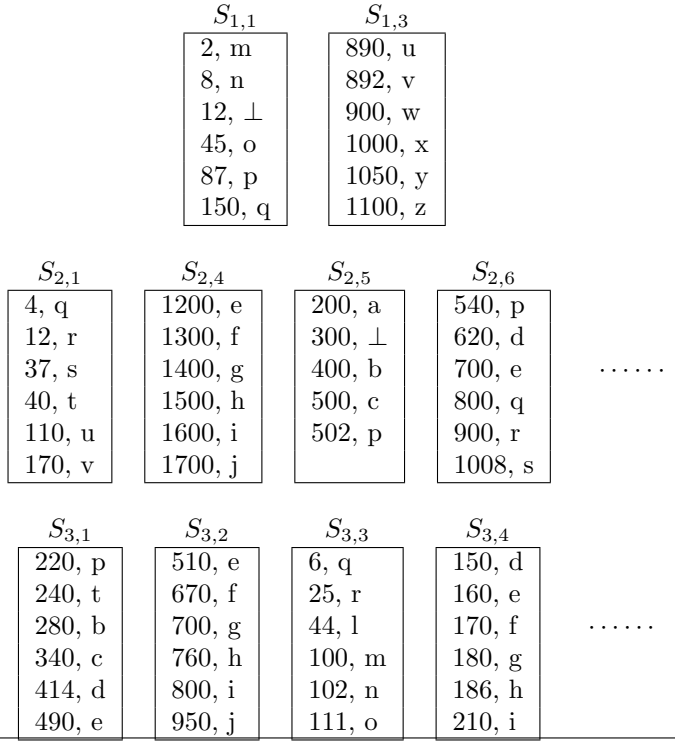
$S_{2,1}$
4, q
8, u
12, r
37, s
39, z
40, t
67, c
110, u
148, x
170, v
200, a
211, b
237, i

$S_{2,2}$
2, m
8, n
9, u
12, ⊥
20, a
21, v
37, w
38, x
87, p
94, b
102, y
110, ⊥
133, d
150, q
220, e

2. Consider the following partially shown LSM storage for a relation that is compacted using LCS. Assume that each SSTable can store at most 6 records and F = 2. Show the state of the LSM storage after the compaction of SSTable $S_{1,2}$.



Solution:



3. Consider storing a relation R using LSM storage that is compacted with LCS and a compaction factor of F. If the maximum number of records in R is n and the size of each record is m MB, how many levels in the LSM storage are required to store R?

> **Solution:** The maximum size of R is mn MB. Let L denote the number of required levels to store R. In the worst case, the last level of the LSM storage stores a version of each record in R. Therefore, $F^{L-1} < mn \leq F^L$. Thus, $log_F(mn) \leq L < log_F(mn) + 1$. That is, $L = \lceil log_F(mn) \rceil$.

4. Consider LSM storage based on LCS with compaction factor F. What's the effect of increasing F on search performance? What's the effect of increasing F on the I/O cost of compaction?

> **Solution:** Increasing F reduces the number of levels of LSM storage which improves on the worst-case I/O cost for searching. However, since F affects the maximum number of overlapping SSTables to be merged during compaction, a larger F increases the I/O cost of compaction as more SSTables will be merged during compaction.

5. Consider the following distributed database scenario based on DynamoDB.

   - The table Customers(region, custId, email, category) has a composite primary key (region, custId) and its partitions are stored in 10 servers. Here, `custId` refers to the customer's unique identifier, `region` refers to the customer's geographical region, `email` refers to the customer's email address, and `category` refers to the customer's status (e.g., Gold, Diamond).
   - There are two secondary indexes on Customers:
     - A local index I_{lsi} with schema (region, category, custId)
     - A global index I_{gsi} with schema (category, region, custId)
   - Assume that each server maintains the storage & index metadata (i.e., hashing functions for partitioning Customers & I_{gsi}) so that it can determine which server is storing (1) the data records with a given `region` value and (2) the I_{gsi} index record with a given `category` value.

   For each of the following queries, determine whether it is more efficient to evaluate the query using either I_{lsi} or I_{gsi}.

   (a) Query Q_1: SELECT * FROM Customers WHERE category = "Gold"

   (b) Query Q_2: SELECT * FROM Customers WHERE region = "Asia" AND category = "Gold"

> **Solution:**
>
> (a) Using I_{lsi} requires searching the index on each of the 10 servers and then retrieving the matching data records from these servers.
>
> Using I_{gsi} requires searching the index on one server (say server S_i) associated with category = "Gold". If the matching data records were distributed on all 10 servers, using I_{gsi} would be less efficient since S_i needs to incur communication latency to send the matching (region, custIds) to the other 9 servers to retrieve the matching data records. On the other hand, if the matching data records were distributed on very few servers, using I_{gsi} could be more efficent since there are fewer servers involved in the query evaluation.
>
> (b) Using I_{lsi} requires searching the index on the server which contains records for region = "Asia" and then retrieving the matching data records (with category = "Gold") from that same server. Thus, only one server is involved in the query evaluation.
>
> Using I_{gsi} requires searching the index on the server (say server S_i) associated with category = "Gold" and then retrieving those matching data records from the server S_j which contains records for region = "Asia". Since S_i and S_j could be different servers, using I_{lsi} should be more efficient.