Harshitha Onkar
Sac Id: 220262706

Bag Difference (R – $_B$ S) – One Pass Algorithm

Step 1: Initial Disk Representation of R and S

Disk Structure for R (r_disk): ArrayList<List<List<Integer>>>, Total no. of Blocks: 4

Disk Structure for S (s_disk): ArrayList<List<List<Integer>>>, Total no. of Blocks: 3

Following is the structural information of R and S.

	R (X, Y, Z)	S(X, Y, Z)
No. of Elements per tuple	3	3
No. of tuples per block	3	3
Total no. of blocks	4	3

Step 2: Loading S into Memory

Memory Structure for S (s_mem): HashMap<List<Integer>, Long> with key as the entire tuple and count as value. Here the search key is the entire tuple.

S is loaded into HashMap structure block by block, and the search structure is updated with tuple and no. of occurrences for the tuple seen so far.

Step 3: Loading R into Memory

Memory Structure for R (r_mem): ArrayList<List<Integer>>, Size: 1 block, 3 tuples/block

Memory Structure for output buffer (out_buffer): ArrayList<List<Integer>>, Size: 1 block, 3 tuples/block

Final Disk Structure (disk): ArrayList<List<List<Integer>>>

R is loaded one block at a time. For each block of R, the following is done:

For each tuple t in block of R the following is done:

- If t is not present in S, then it is added to output buffer
- Else if t is present in S and count == 0 then t is added to output buffer
- Else if t is present in S and count > 0 then count is reduced by 1 and t is skipped

When output buffer is full, (size == block size) then the contents of output buffer is dumped into disk and the output buffer is cleared.

After processing a block of R, r_mem is cleared, and the next block is loaded into r_mem

Step 4: Final Contents of the disk

Any remaining contents in the output buffer is dumped onto the disk
Input:

R [[[1, 2, 3], [2, 2, 2], [2, 4, 8]], [[2, 2, 2], [2, 2, 2], [1, 1, 1]], [[5, 5, 5], [2, 2, 2], [1, 1, 1]], [[5, 5, 5], [1, 1, 1], [3, 3, 3]]]
S [[[2, 2, 2], [2, 2, 2], [3, 3, 3]], [[1, 1, 1], [2, 2, 2], [1, 1, 1]], [[4, 4, 4], [2, 2, 2], [9, 9, 9]]]

Final Output: [[1, 2, 3], [2, 4, 8], [5, 5, 5], [5, 5, 5], [1, 1, 1]]

Harshitha Onkar
Sac Id: 220262706

Execution Results

```
------------------------------ STEP 1: R and S on Disk ----------------------------

R ---> [[[1, 2, 3], [2, 2, 2], [2, 4, 8]], [[2, 2, 2], [2, 2, 2], [1, 1, 1]], [[5, 5,
5], [2, 2, 2], [1, 1, 1]], [[5, 5, 5], [1, 1, 1], [3, 3, 3]]]

S ---> [[[2, 2, 2], [2, 2, 2], [3, 3, 3]], [[1, 1, 1], [2, 2, 2], [1, 1, 1]], [[4, 4,
4], [2, 2, 2], [9, 9, 9]]]

--------------------------- STEP 2: Loading S into Memory -----------------------
Loading S into Memory (block by block)

After Block 1:{[2, 2, 2]=2, [3, 3, 3]=1}
After Block 2:{[1, 1, 1]=2, [2, 2, 2]=3, [3, 3, 3]=1}
After Block 3:{[1, 1, 1]=2, [2, 2, 2]=4, [3, 3, 3]=1, [4, 4, 4]=1, [9, 9, 9]=1}

STEP 3: Loading R into Memory (block by block) and comparing each tuple in R with S

After loading Block_1 of R into memory
[[1, 2, 3], [2, 2, 2], [2, 4, 8]]

After loading Block_2 of R into memory
[[2, 2, 2], [2, 2, 2], [1, 1, 1]]

After loading Block_3 of R into memory
[[5, 5, 5], [2, 2, 2], [1, 1, 1]]

Output buffer and Disk Contents: BEFORE TRANSFER

Output buffer: [[1, 2, 3], [2, 4, 8], [5, 5, 5]]
Disk: []

Output buffer and Disk Contents: AFTER TRANSFER

Output buffer: []
Disk: [[1, 2, 3], [2, 4, 8], [5, 5, 5]]

After loading Block_4 of R into memory
[[5, 5, 5], [1, 1, 1], [3, 3, 3]]

----------- STEP 4: Transferring any Remaining tuples in buffer to disk -------------

Remaining tuples at the end: BEFORE TRANSFER

Output buffer: [[5, 5, 5], [1, 1, 1]]
Disk: [[1, 2, 3], [2, 4, 8], [5, 5, 5]]

Remaining tuples at the end: AFTER TRANSFER

Output buffer: []
Disk: [[1, 2, 3], [2, 4, 8], [5, 5, 5], [5, 5, 5], [1, 1, 1]]

--------------------------- STEP 5: Final Disk Contents ---------------------------
Final Result on Disk: [[1, 2, 3], [2, 4, 8], [5, 5, 5], [5, 5, 5], [1, 1, 1]]
```