SNS COLLEGE OF TECHNOLOGY

COIMBATORE

AN AUTONOMOUS INSTITUTION

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE New Delhi & affiliated to the Anna University, Chennai

DEPARTMENT OF MCA

Course Name : 19CAT609 - DATA BASE MANAGEMENT SYSTEM

Class : I Year / II Semester

Unit IV - QUERY EVALUATION AND DATABASE DESIGN

Topic II – External sorting

Sorting

•We may build an index on the relation, and then use the index to read the relation in sorted order. May lead to one disk block access for each tuple.

•For relations that fit in memory, techniques like quicksort can be used. For relations that don't fit in memory, **external sort-merge** is a good choice.

2

External Sort-Merge

Let *M* denote memory size (in pages).

1. **Create sorted runs**. Let *i* be 0 initially.

   Repeatedly do the following till the end of the relation:

   (a) Read *M* blocks of relation into memory

   (b) Sort the in-memory blocks

   (c) Write sorted data to run R_i; increment *i.*

   Let the final value of *i* be *N*

2. *Merge the runs (next slide)…..*

3

External Sort-Merge

2.Merge the runs (N-way merge). We assume (for now) that $N < M$.

1. Use N blocks of memory to buffer input runs, and 1 block to buffer output. Read the first block of each run into its buffer page
2. **repeat**
   1. Select the first record (in sort order) among all buffer pages
   2. Write the record to the output buffer. If the output buffer is full write it to disk.
   3. Delete the record from its input buffer page.
      If the buffer page becomes empty **then**
         read the next block (if any) of the run into the buffer.
3. **until** all input buffer pages are empty:

External Sort-Merge

If $N \geq M$, several merge *passes* are required.

In each pass, contiguous groups of M - 1 runs are merged.

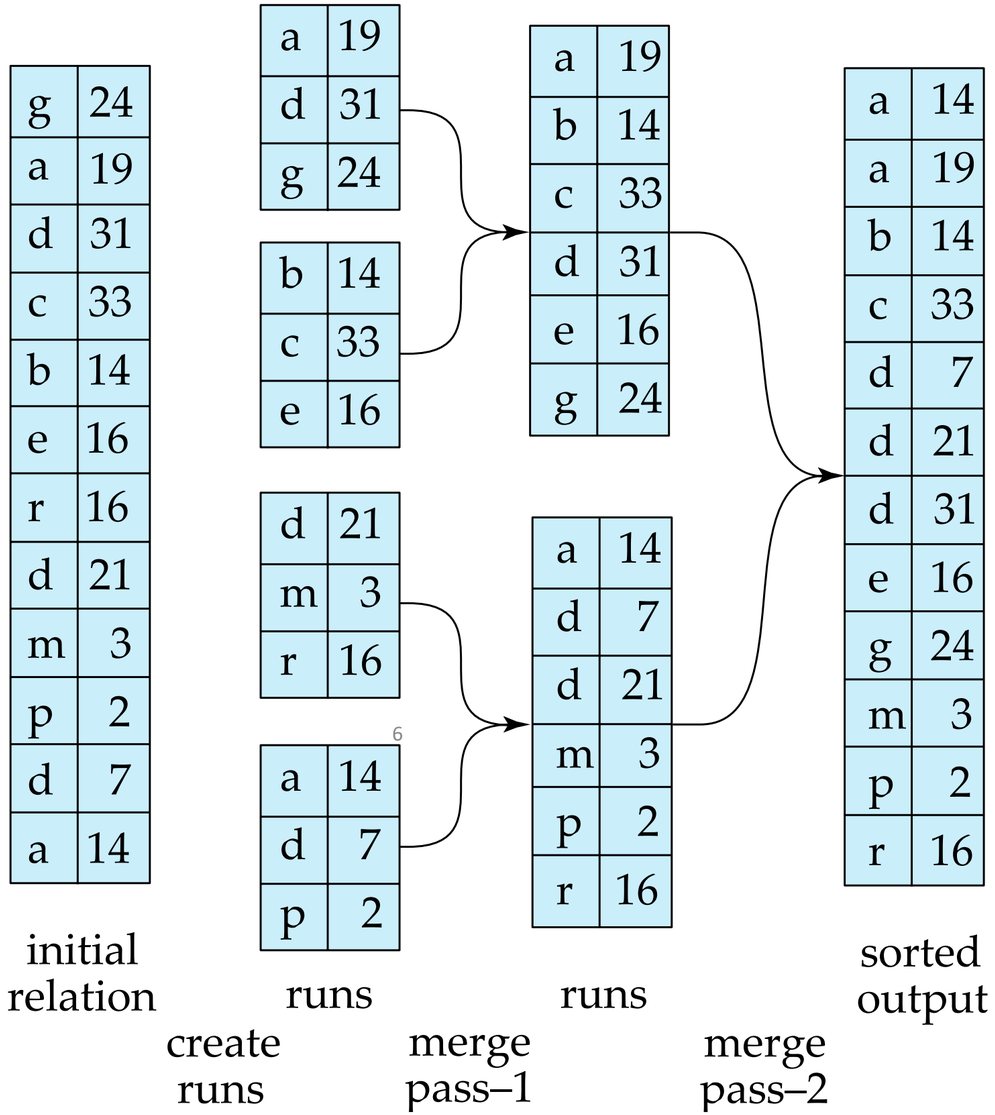
A pass reduces the number of runs by a factor of M -1, and creates runs longer by the same factor.

E.g. If M=11, and there are 90 runs, one pass reduces the number of runs to 9, each 10 times the size of the initial runs

5

Repeated passes are performed till all runs have been merged into one.

Example: External Sorting Using Sort-Merge



External Merge Sort

Cost analysis:

1 block per run leads to too many seeks during merge

Instead use b_b buffer blocks per run

➔ read/write b_b blocks at a time

Can merge $\lfloor M/b_b \rfloor - 1$ runs in one pass

Total number of merge passes required: $\lceil \log_{\lfloor M/bb \rfloor - 1}(b_r/M) \rceil$.

Block transfers for initial run creation as well as in each pass is $2b_r$

for final pass, we don't count write cost

we ignore final write cost for all operations since the output of an operation may be sent to the parent operation without being written to disk

7

Thus total number of block transfers for external sorting:

$$b_r (2 \lceil \log_{\lfloor M/bb \rfloor - 1} (b_r / M) \rceil + 1) \lceil$$

Seeks: next slide

External Merge Sort (Cont.)

Cost of seeks

During run generation: one seek to read each run and one seek to write each run

$2\lceil b_r / M \rceil$

During the merge phase

Need $2\lceil b_r / b_b \rceil$ seeks for each merge pass

except the final one which does not require a write

Total number of seeks:

$2\lceil b_r / M \rceil + \lceil b_r / b_b \rceil (2\lceil \log_{\lfloor M/bb \rfloor - 1}(b_r / M) \rceil - 1)$

Reference

1. https://www.tutorialspoint.com/dbms/dbms_file_structure.htm#:~:text=Relative%20data%20and%20information%20is,blocks%20that%20can%20store%20records.
2. https://www.javatpoint.com/dbms-file-organization
3. https://www.tutorialspoint.com/dbms/dbms_storage_system.htm

THANK YOU