Query Processing 2

Instructor: Beom Heyn Kim

beomheynkim@hanyang.ac.kr

Department of Computer Science

Overview

- Sorting
- Assignments

Sorting

- Sorting is important in database systems:
  - The output of SQL queries may need to be sorted
  - Query processing is efficient if the input relations are sorted
- Sorting approaches:
  - Build an index on the sort key, and then use the index to read the relation in sorted order.
    - May lead to one disk block access for each record.
      - Note: There are going to be much larger number of records than the number of blocks.
  - For relations that fit in memory, techniques like quicksort can be used.
    - However, relations are often too large to fit entirely in memory
  - For relations that don't fit in memory, **external sort-merge** is a good choice

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)…..*

External Sort-Merge (Cont.)

2. **Merge the runs (N-way merge)**. We assume (for now) that $N < M$.
   a. 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
   b. **Repeat**
      i. Select the first record (in sort order) among all buffer pages
      ii. Write the record to the output buffer. If the output buffer is full write it to disk.
      iii. 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.
   c. **until** all input buffer pages are empty:

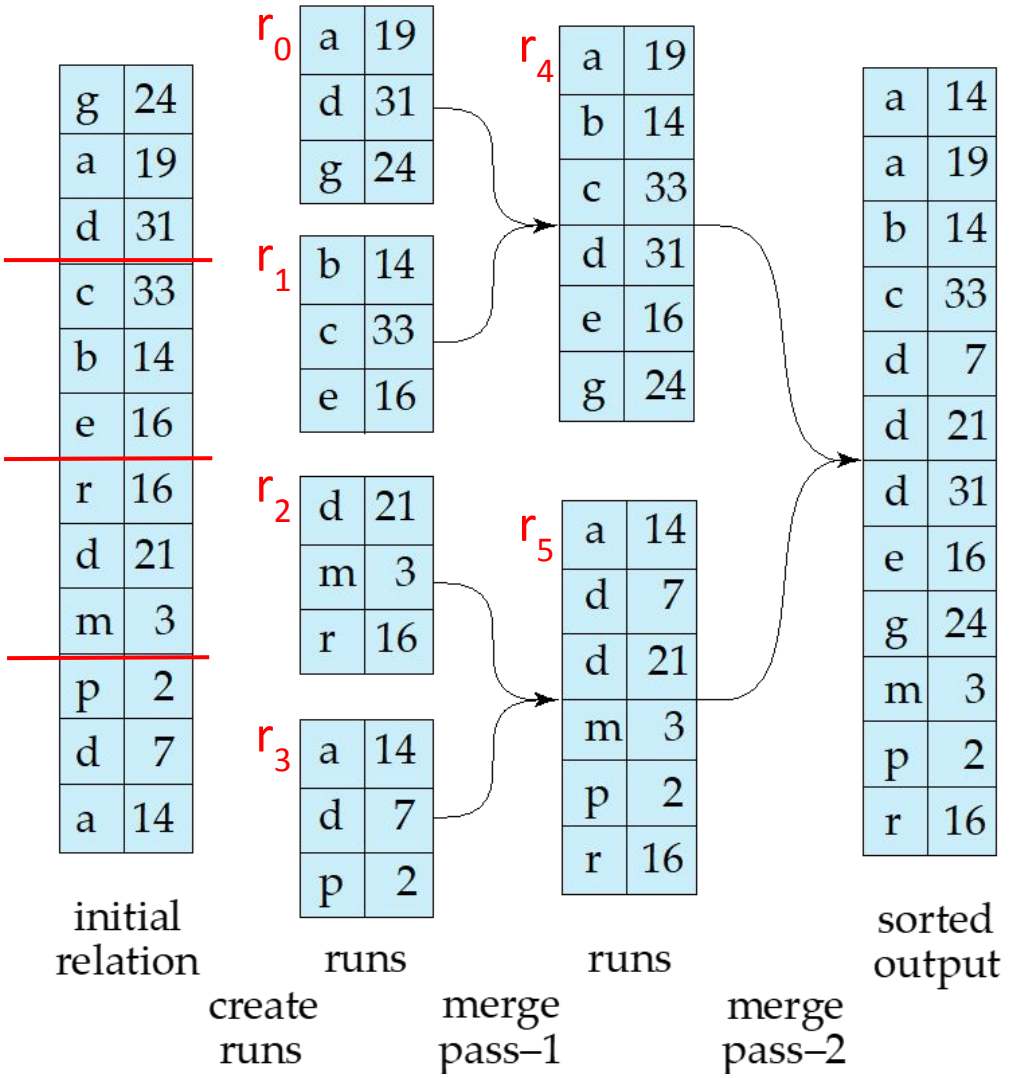
External Sort-Merge (Cont.)

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

Example: External Sorting Using Sort-Merge

Assume:

1. 3 blocks available in memory
2. 1 tuple fits in each block
3. During the first stage where the runs are created, all blocks in memory can be used as buffers for both input and output
4. During the merge passes, 2 blocks are used as input buffers and 1 block is used as the output buffer



External Merge Sort (Cont.)

- 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: $\lceil \log_{\lfloor M/b_b \rfloor - 1}(b_r / M) \rceil$.
    - b_r denotes the number of blocks containing records of relation r
  - Block transfers for initial run creation is $2b_r$
    - The first stage reads every block of the relation and writes them out again
  - For each merge pass, there will be $2b_r$ block transfers as well
    - 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
    - Thus total number of block transfers for external sorting:

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

  - 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:

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

Overview

- Sorting
- Assignments

Assignments

- Reading: Ch15.4
- Practice Excercises: 15.1

Solutions to the Practice Excercises:
https://www.db-book.com/Practice-Exercises/index-solu.html

The End