SC3020-CZ4031: Tutorial 6

Classroom Discussion

Problem 1. Suppose that we have three sets of integers, denoted as S_1, S_2, and S_3, respectively. Each set is sorted and given to you in a file. Assume that each disk block can hold 2 integers, and the memory has 4 blocks. The content of S_1, S_2, and S_3 is shown below.

S_1 : $[1, 10], [20, 30], [40, 50]$
S_2 : $[5, 35], [37, 38]$
S_3 : $[45, 60]$

The notation $[x, y]$ represents a block holding integers x and y. It is clear that S_1, S_2, and S_3 occupy 3, 2, and 1 block, respectively.

Demonstrate how to merge S_1, S_2, and S_3 into one sorted file using the merging algorithm taught in the class. What is the I/O cost of the algorithm?

Solution. The merging algorithm allocates (i) one memory block as the input buffer to read S_i, for each $i \in [1, 3]$, and (ii) one memory block as the output buffer. In the beginning, the algorithm loads the first block of each input file into memory, whose content is shown below:

Memory: $[1, 10], [5, 35], [45, 60], [\quad]$

As long as no input buffer is empty, the algorithm moves the smallest integer in the 3 input buffers to the output buffer. This yields:

Memory: $[\quad, 10], [\quad, 35], [45, 60], [1, 5]$

Now the output buffer is full and thus flushed to the disk:

Memory: $[\quad, 10], [\quad, 35], [45, 60], [\quad]$
Output file on disk: $[1, 5]$

Moving the next smallest element in the input buffers to the output buffer yields:

Memory: $[\quad], [\quad, 35], [45, 60], [10, \quad]$

The first input buffer is full, prompting the algorithm to read the next page of S_1:

Memory: $[20, 30], [\quad, 35], [45, 60], [10, \quad]$

The next few steps of the algorithm are straightforward:

Memory: $[\quad, 30], [\quad, 35], [45, 60], [10, 20]$
$\Rightarrow$
Memory: $[\quad, 30], [\quad, 35], [45, 60], [\quad]$
Output file on disk: $[1, 5], [10, 20]$
$\Rightarrow$
Memory: $[\quad], [\quad, 35], [45, 60], [30, \quad]$

Reading the next page of S_1 gives:

Memory: $\underline{[40,50]}, \underline{[\quad\quad]}, \underline{[45,60]}, \underline{[30,35]}$

$\Rightarrow$

Memory: $\underline{[40,50]}, \underline{[\quad\quad]}, \underline{[45,60]}, \underline{[\quad\quad]}$

Output file on disk: $\underline{[1,5]}, \underline{[10,20]}, \underline{[30,35]}$

Now we reading the next page of S_2:

Memory: $\underline{[40,50]}, \underline{[37,38]}, \underline{[45,60]}, \underline{[\quad\quad]}$

$\Rightarrow$

Memory: $\underline{[40,50]}, \underline{[\quad\quad]}, \underline{[45,60]}, \underline{[37,38]}$

$\Rightarrow$

Memory: $\underline{[40,50]}, \underline{[\quad\quad]}, \underline{[45,60]}, \underline{[\quad\quad]}$

Output file on disk: $\underline{[1,5]}, \underline{[10,20]}, \underline{[30,35]}, \underline{[37,38]}$

No need to replenish the input buffer of S_2 because this file has been exhausted. The remaining execution should be straightforward:

Memory: $\underline{[\quad,50]}, \underline{[\quad\quad]}, \underline{[\quad,60]}, \underline{[40,45]}$

$\Rightarrow$

Memory: $\underline{[\quad,50]}, \underline{[\quad\quad]}, \underline{[\quad,60]}, \underline{[\quad\quad]}$

Output file on disk: $\underline{[1,5]}, \underline{[10,20]}, \underline{[30,35]}, \underline{[37,38]}, \underline{[40,45]}$

$\Rightarrow$

Memory: $\underline{[\quad\quad]}, \underline{[\quad\quad]}, \underline{[\quad\quad]}, \underline{[50,60]}$

$\Rightarrow$

Memory: $\underline{[\quad\quad]}, \underline{[\quad\quad]}, \underline{[\quad\quad]}, \underline{[\quad\quad]}$

Output file on disk: $\underline{[1,5]}, \underline{[10,20]}, \underline{[30,35]}, \underline{[37,38]}, \underline{[40,45]}, \underline{[50,60]}$

The total I/O cost is 12 because every block in the input files is read once and every block in the output file is written once.

Problem 2. Assume that each disk block can hold 2 integers, and the memory has 3 blocks. You are given a set S stored on the disk in 9 blocks as shown below:

S : $\underline{[50,30]}, \underline{[80,20]}, \underline{[10,70]}, \underline{[40,60]}, \underline{[55,25]}, \underline{[90,5]}, \underline{[85,95]}, \underline{[35,15]}, \underline{[65,75]}$

Suppose that we execute the initial step of the external sort algorithm taught in the class. Show all the sorted runs produced by this step. What is the I/O cost of this step?

Solution. The initial step reads 3 blocks of S into memory at a time, sort them (in memory), and write the sorted list into a sorted run. Therefore, the first sorted run is:

$$\underline{[10,20]}, \underline{[30,50]}, \underline{[70,80]},$$

the second sorted run is

$$\underline{[5,25]}, \underline{[40,55]}, \underline{[60,90]},$$

and the last sorted run is

$$\underline{[15,35]}, \underline{[60,75]}, \underline{[85,95]}.$$

The I/O cost is 18 (9 read I/Os + 9 write I/Os).

Problem 3. Continuing on Problem 2, now execute the first merging step on the sorted runs you obtained. Show the sorted runs at the end of this step. What is the I/O cost of this step?

Solution. The merging step combines two sorted runs from the previous step into a single new sorted run at a time. Hence, the first new sorted run is:

$$\underline{[5, 10]}, \underline{[20, 25]}, \underline{[30, 40]}, \underline{[50, 55]}, \underline{[60, 70]}, \underline{[80, 90]}.$$

The 3rd sorted from the previous step has no other sorted run to be combined with. Hence, it is taken as the second new sorted run directly:

$$\underline{[15, 35]}, \underline{[60, 75]}, \underline{[85, 95]}.$$

The total number of I/Os is 12 (6 read I/Os + 6 write I/Os).

Problem 4. Let S be a set of integers stored in 100000 blocks. If the memory has $M = 100$ blocks. How many merging steps are required to sort S?

Solution. The initial step creates $100000/100 = 1000$ sorted runs. Each merging step combines $M - 1 = 99$ sorted runs into a single new sorted run. Therefore, there are $\lceil 1000/99 \rceil = 11$ sorted runs after the first merging step. Another merging step will complete the sort. The answer is therefore 2.

Problem 5. Let S be a set of integers stored in 10^9 blocks. If we sort S using the external sort algorithm, what is the smallest number M of memory blocks required for the algorithm to incur only one merging step?

Solution. The initial step creates $\lceil 10^9/M \rceil$ sorted runs. If sorting needs to be completed with only one merging step, we need

$$\lceil 10^9/M \rceil \leq M - 1.$$

Solving the inequality for integer M yields $M \geq 31624$.

Critical Thinking

Problem 6 Let $R(A, B)$ be a relation with attributes A and B. No two tuples in R are identical. Explain how to sort R according to the following order: rank tuple t_1 before tuple t_2 if (i) $t_1.A < t_2.A$ or (ii) $t_1.A = t_2.A$ and $t_1.B < t_2.B$.

Problem 7 Let S be a *bag* (a.k.a. multi-set) of integers, namely, S may contain duplicate integers. Adapt the external sort algorithm to sort S.

Problem 8 Let S be a *bag* (a.k.a. multi-set) of integers, namely, S may contain duplicate integers. Adapt the external sort algorithm to output a disk file containing the *distinct* integers of S in ascending order.

Problem 9 Discuss how to process the following query on a relation $R(A, B)$:

`SELECT` A`, MAX(`B`) FROM` R `GROUP BY` A