CARNEGIE MELLON UNIVERSITY
COMPUTER SCIENCE DEPARTMENT
15-445/645 – DATABASE SYSTEMS (FALL 2021)
PROF. LIN MA

Homework #3 (by Sophie Qiu) – Solutions

Due: **Sunday Oct 24, 2021 @ 11:59pm**

IMPORTANT:

- **Upload this PDF** with your answers to **Gradescope by 11:59pm on Sunday Oct 24, 2021**.
- **Plagiarism**: Homework may be discussed with other students, but all homework is to be completed **individually**.
- **You have to use this PDF for all of your answers.**

For your information:

- Graded out of **100** points; **2** questions total
- Rough time estimate: ≈ 1 - 2 hours (0.5 - 1 hours for each question)

Revision : 2021/11/16 01:04

Question	Points	Score
Sorting Algorithms	40	
Join Algorithms	60	
Total:	100	

Question 1: Sorting Algorithms . . . . [40 points]

Graded by:

We have a database file with six million pages (N = 6,000,000 pages), and we want to sort it using external merge sort. Assume that the DBMS is not using double buffering or blocked I/O, and that it uses quicksort for in-memory sorting. Let B denote the number of buffers.

(a) **[10 points]** Assume that the DBMS has six buffers. How many passes does the DBMS need to perform in order to sort the file?

☐ 5 ☐ 7 ☐ 8 ■ **10** ☐ 12

Solution:

$$\begin{aligned} 1+\left\lceil \log_{B-1}\left(\left\lceil \frac{N}{B}\right\rceil\right)\right\rceil &= 1+\lceil \log_5(\lceil 6,000,000/6\rceil)\rceil \\ &= 1+\lceil \log_5(\lceil 1,000,000\rceil)\rceil \\ &= 1+9=10 \end{aligned}$$

(b) **[5 points]** Again, assuming that the DBMS has six buffers. What is the total I/O cost to sort the file?

☐ 60,000,000 ■ **120,000,000** ☐ 144,000,000 ☐ 240,000,000 ☐ 480,000,000

Solution: $Cost = 2N \times \#passes = 2 \times 6,000,000 \times 10$

(c) **[10 points]** What is the smallest number of buffers B that the DBMS can sort the target file using only two passes?

☐ 172 ☐ 173 ☐ 174 ■ **2,450** ☐ 2,451 ☐ 2,452 ☐ 2,827 ☐ 2,828 ☐ 2,829 ☐ 3,999,999 ☐ 4,000,000 ☐ 4,000,001

Solution: We want B where $N \le B \times (B-1)$. If $B = 2450$, then $6,000,000 \le 2050 \times 2449 = 6,000,050$; any smaller value for B would fail.

(d) **[10 points]** What is the smallest number of buffers B that the DBMS can sort the target file using only six passes?

☐ 14 ■ **15** ☐ 16 ☐ 1,240 ☐ 1,241 ☐ 1,242 ☐ 1,256 ☐ 1,257 ☐ 1,258 ☐ 2,934 ☐ 2,935 ☐ 2,936 ☐ 3,999,999 ☐ 4,000,000 ☐ 4,000,001

Solution: $B \times (B-1)^5 = 15 \times 14 \times 14 \times 14 \times 14 \times 14 = 8,067,360$. Any smaller value of B would fail.

(e) **[5 points]** Suppose the DBMS has twenty-four buffers. What is the largest database file (expressed in terms of N, the number of pages) that can be sorted with external merge sort using six passes?

☐ 65,610 ☐ 65,601 ☐ 131,071 ☐ 131,072 ☐ 3,590,490 ☐ 3,590,940
☐ 49,251,980 ☐ 49,521,980 ☐ 154,472,230 ■ **154,472,232**

Solution: We want N such that $N \leq B \times (B-1)^5$. The largest such value is $B \times (B-1)^5$ itself, which is $24 \times 23^5 = 154,472,232$

Question 2: Join Algorithms .... [60 points]

Graded by:

Consider relations `R(a, b)`, `S(a, c, d)`, and `T(a, e)` to be joined on the common attribute `a`. Assume that there are no indexes available on the tables to speed up the join algorithms.

- There are B = 60 pages in the buffer
- Table `R` spans M = 1,400 pages with 60 tuples per page
- Table `S` spans N = 2,200 pages with 200 tuples per page
- The joining result of `R` and `S` spans K = 2,000 pages
- Table `T` spans L = 1,000 pages with 200 tuples per page

Answer the following questions on computing the I/O costs for the joins. You can assume the simplest cost model where pages are read and written one at a time. You can also assume that you will need one buffer block to hold the evolving output block and one input block to hold the current input block of the inner relation. You may ignore the cost of the writing of the final results.

(a) **[5 points]** Block nested loop join with `R` as the outer relation and `S` as the inner relation:
☐ 11,200 ☐ 23,000 ■ **56,400** ☐ 85,000 ☐ 92,600

Solution: $M + \lceil \frac{M}{B-2} \rceil \times N = 1,400 + \lceil \frac{1,400}{58} \rceil \times 2,200 = 1,400 + 55,000 = 56,400$

(b) **[5 points]** Block nested loop join with `S` as the outer relation and `R` as the inner relation:
☐ 31,200 ☐ 43,000 ☐ 43,600 ☐ 52,900 ■ **55,400**

Solution: $N + \lceil \frac{N}{B-2} \rceil \times M = 2,200 + \lceil \frac{2,200}{58} \rceil \times 1,400 = 2,200 + 53,200 = 55,400$

(c) Hash join with `S` as the outer relation and `R` as the inner relation. You may ignore recursive partitioning and partially filled blocks.

i. **[5 points]** What is the cost of the partition phase?
☐ 2,800 ☐ 4,400 ☐ 5,000 ☐ 5,800 ■ **7,200**

Solution: $2 \times (M + N) = 2 \times (1,400 + 2,200) = 2 \times 3,600 = 7,200$

ii. **[5 points]** What is the cost of the probe phase?
☐ 2,800 ☐ 4,400 ■ **3,600** ☐ 4,800 ☐ 7,200

Solution: $(M + N) = (1,400 + 2,200) = 3,600$

(d) **[10 points]** Assume that the tables do not fit in main memory and that a high cardinality of distinct values hash to the same bucket using your hash function h_1. Which of the following approaches works the best?

☐ Create hashtables for the inner and outer relation using h_1 and rehash into an embedded hash table using h_1 for large buckets

■ **Create hashtables for the inner and outer relation using h_1 and rehash into an embedded hash table using h_2 != h_1 for large buckets**

☐ Use linear probing for collisions and page in and out parts of the hashtable needed at a given time

☐ Create 2 hashtables half the size of the original one, run the same hash join algorithm on the tables, and then merge the hashtables together

Solution: Use Grace hash join with recursive partitioning, which is what the correct option describes.

(e) Sort-merge join with S as the outer relation and R as the inner relation:

i. **[4 points]** What is the cost of sorting the tuples in R on attribute a?
☐ 3,000 ■ **5,600** ☐ 7,400 ☐ 9,600 ☐ 10,800

Solution: $passes = 1 + \lceil \log_{B-1}(\lceil \frac{M}{B} \rceil) \rceil = 1 + \lceil \log_{59}(\lceil \frac{1{,}400}{60} \rceil) \rceil = 1 + 1 = 2$
$2M \times passes = 2 * 1,400 * 2 = 5,600$

ii. **[4 points]** What is the cost of sorting the tuples in S on attribute a?
☐ 3,400 ☐ 4,000 ☐ 6,400 ☐ 7,600 ■ **8,800**

Solution: $passes = 1 + \lceil \log_{B-1}(\lceil \frac{N}{B} \rceil) \rceil = 2$
$2N \times passes = 2 * 2,200 * 2 = 8,800$

iii. **[10 points]** What is the cost of the merge phase assuming there are no duplicates in the join attribute?
☐ 1,400 ☐ 1,800 ■ **3,600** ☐ 4,400 ☐ 4,800

Solution: $M + N = 1,400 + 2,200 = 3,600$

iv. **[10 points]** What is the cost of the merge phase in the worst-case scenario?
☐ 1,080,000 ☐ 2,880,000 ■ **3, 080,000** ☐ 4, 750,000 ☐ 10,080,000

Solution: $M \times N = 1,400 \times 2,200 = 3,080,000$

v. **[2 points]** Now consider joining R, S and then joining the result with T. What is the cost of the merge phase assuming there are no duplicates in the join attribute?
☐ 1,000 ☐ 2,000 ■ **3,000** ☐ 5,000 ☐ 2,000,000

Solution: $K + L = 2,000 + 1,000 = 3,000$