CARNEGIE MELLON UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE
15-445/645 – DATABASE SYSTEMS (FALL 2018)
PROF. ANDY PAVLO

Homework 3 (by Lin Ma)
Due: **Monday Oct 15, 2018 @ 11:59pm**

IMPORTANT:

- **Upload this PDF** with your answers to **Gradescope by 11:59pm on Monday Oct 15, 2018**.
- **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: $\approx$ 1 - 2 hours (0.5 - 1 hours for each question)

Revision : 2018/10/08 18:37

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

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

We have a database file with a million pages (N = 4,000,000 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?
□ 4 □ 6 □ 8 □ 10 □ 12

(b) **[10 points]** Again, assuming that the DBMS has six buffers. What is the total I/O cost to sort the file?
□ 56,000,000 □ 64,000,000 □ 80,000,000 □ 40,000,000 □ 88,000,000

(c) **[10 points]** What is the smallest number of buffers B that the DBMS can sort the target file using only two passes?
□ 44 □ 45 □ 46 □ 158 □ 159 □ 160 □ 161 □ 1,999 □ 2,000
□ 2,001 □ 4,000,000 □ 4,000,001

(d) **[10 points]** What is the smallest number of buffers B that the DBMS can sort the target file using only three passes?
□ 44 □ 45 □ 46 □ 158 □ 159 □ 160 □ 161 □ 1,999 □ 2,000
□ 2,001 □ 4,000,000 □ 4,000,001

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

Consider relations `R(a, b)` and `S(a, c, d)` to be joined on the common attribute `a`. Assume that there are no indexes.

- There are B = 50 pages in the buffer
- Table `R` spans M = 2000 pages with 80 tuples per page
- Table `S` spans N = 300 pages with 40 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 final writing of the results.

(a) **[10 points]** Block nested loop join with `R` as the outer relation and `S` as the inner relation:
□ 12,000 □ 12,300 □ 12,600 □ 14,300 □ 14,600

(b) **[5 points]** Block nested loop join with `S` as the outer relation and `R` as the inner relation:
□ 12,000 □ 12,300 □ 14,000 □ 14,300 □ 16,300

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

i. **[10 points]** What is the cost of sorting the tuples in `R` on attribute `a`?
□ 7,854 □ 7,772 □ 5,833 □ 1,166 □ 875

ii. **[5 points]** What is the cost of sorting the tuples in `S` on attribute `a`?
□ 7,854 □ 7,772 □ 5,833 □ 1,166 □ 875

iii. **[10 points]** What is the cost of the merge phase assuming there are no duplicates in the join attribute?
□ 2,300 □ 4,600 □ 6,900 □ 154 □ 77

iv. **[10 points]** What is the cost of the merge phase in the worst case scenario?
□ 2,300 □ 6,900 □ 600,000 □ 1,200,000 □ 300,000,000

(d) 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?
□ 2,300 □ 4,600 □ 6,900 □ 3,600 □ 1,000

ii. **[5 points]** What is the cost of the probe phase?
□ 2,300 □ 4,600 □ 6,900 □ 3,600 □ 1,000