The Chinese University of Hong Kong
Department of Systems Engineering and Engineering Management

SEEM3550A Fundamentals of Information Systems

2021-22 Assignment 3: Query Processing

This is an individual assignment. Everyone must submit it individually. The due date for the assignment 3 is 17:00, Apr 29th, 2022. The late penalty will be 10% per day. A submission will not be accepted since the fifth day from the deadline.

Q1. **[19 marks]** Answer the following questions related to query cost.

(a) **[3 marks]** Does a relational algebra expression specify a unique evaluation algorithm?

(b) **[3 marks]** When computing the cost of a query execution, do we include the cost of writing output to disk in the cost formulae? Explain briefly.

(c) **[3 marks]** "For any query, the number of disk seeks involved must be less than or equal to that of block transfers"? Is this statement true? Explain briefly.

(d) **[4 marks]** Evaluating an SQL query a on an instance of relations outputs O_a as its result. Given another SQL query b, we find that its output $O_b = O_a$ on the same instance of relations and that the query cost (as measured by the number of block transfers and disk seeks) of b is less than that of a. Please suggest two possible reasons why we may not want to replace a with b.

(e) **[2 marks]** During a join operation, at least how many blocks have to be allocated to the memory buffer? Explain briefly.

(f) **[4 marks]** Let r and s denote two relations, n_r and n_s denote the number of records in relation r and relation s, and b_r and b_s denote the number of blocks used by relation r and relation s. What is the minimum value for the memory buffer M to achieve the best case in terms of the join cost? Can we further reduce the cost of joining r and s if we further increase the memory buffer size? Explain briefly.

Q2. **[62 marks]** You are given the following three relations in a university database as below:

COURSE (Cid, Name)

STUDENT (Id, Name, DeptName, GPA)

TAKES (Id, Cid, Semester, Year)

The relation STUDENT has 500 tuples, and 25 tuples fit into one block; the relation COURSE has 200 tuples, and 20 tuples fit into one block; the relation TAKES has 4,000 tuples and 200 tuples fit into one block. The records of STUDENT are sorted in ascending order of Id on disk, and the records of COURSE are sorted in ascending order of Name on disk. Assume a relation is stored on disk consecutively.

(a) **[16 marks]** Answer the following questions and elaborate on your answers.

i. **[4 marks]** What is the implication of the assumption that a relation is stored on disk consecutively?

ii. **[4 marks]** What is a primary index and a secondary index?

iii. **[4 marks]** "A secondary index must be dense for the index to be useful". Is this statement true?

iv. **[4 marks]** It is known that COURSE has a primary index A and a secondary index B. On what search keys are A and B built on respectively?

(b) **[24 marks]** Answer the following questions and show your calculations clearly.

i. **[5 marks]** Consider the following SQL query:

select * from COURSE where Cid=100

Assume COURSE has a B^+-tree index of height 3 on the Cid attribute. Further assume that the Cid of COURSE are integers between 1 to 200. What is the number of disk seeks and the number of block transfers associated with the SQL query in the best case and worst case respectively?

ii. **[5 marks]** Consider the following SQL query:

select * from STUDENT where Id=100

Assume that the Id of STUDENT are integers that are ≥ 1. What is the number of disk seeks and the number of block transfers associated with the SQL query in the best case and worst case respectively?

iii. **[5 marks]** Consider the following SQL query:

select * from STUDENT where Id>100

Assume that the Id of STUDENT are integers between 1 to 500. What is the number of disk seeks and the number of block transfers associated with the SQL query in the best case and worst case respectively?

iv. **[5 marks]** Consider the following SQL query:

select * from STUDENT where Id>100

Assume that the Id of STUDENT are integers between 1 to 500. Further assume that STUDENT has a B+-tree index of height 5 on the Id attribute. If we use the B+-tree for the query, what is the number of disk seeks and the number of block transfers associated with the SQL query in the best case and worst case respectively?

v. **[4 marks]** Is it always better to use a B+-tree than not using one? Please explain your answer by referring to your answers in 2(b)iii and 2(b)iv above.

(c) **[8 marks]** Assume STUDENT has a B+-tree index of height 4 on the DeptName attribute. Further assume there are 25 STUDENTs from the Biology Department.

i. **[4 marks]** Is the B+-tree a primary or secondary index? Why?

ii. **[4 marks]** Consider the following SQL query:

select * from STUDENT where DeptName = 'Biology'

If we use the B+-tree for the query, what is the number of disk seeks and the number of block transfers associated with the SQL query in the worst case?

(d) **[14 marks]** Consider the following SQL query performing join operation:

select * from STUDENT, TAKES where STUDENT.Id = TAKES.Id

i. **[8 marks]** Assume that 22 blocks are allocated to the memory buffer. The query is implemented by nested-loop join. In the best case, which of the relations should be the inner relation? What is the number of disk seeks and the number of block transfers? Explain your answer by comparing the two cases where different relations are chosen as the inner relation.

ii. **[6 marks]** Assume that STUDENT has a B+-tree index of height 6 on the Id attribute. The query is implemented by indexed nested-loop join with TAKES as the outer relation. What is the number of disk seeks and the number of block transfers?

Q3. **[19 marks]** Consider the following two relations R_1 and R_2:

a	b
A	1
B	7
C	2
D	5
A	4
B	3
E	6
F	8

R_1

b	c
1	P
1	Q
2	Q

R_2

We would like to merge-join R_1 and R_2 on attribute b. However, R_1 is unsorted on b, so we have to sort R_1 before joining the relations. The records are stored sequentially from top to bottom as shown. Assume 1 record fits into 1 block of memory. Please show how an external sort-merge on R_1 on the attribute b is conducted, by illustrating the following steps:

(a) **[5 marks]** Create sorted runs, assuming 2 blocks are allocated to the memory buffer.

- Draw figures as in p.35 of Ch. 12 lecture slides
- There should be 4 runs. Label them sequentially as r_1, …, r_4.

(b) **[10 marks]** Merge the sorted runs created above via 4-way merge (p.36 of Ch. 12 lecture slides).

Memory buffer (blocks):

M_1	M_2	M_3	M_4	O

- Show this process by stating the read/write operations in the form of "[Read/Write] [X] to [Y]". For instance "Read r_1 to M_1" stands for reading a block of r_1 from disk to the memory block M_1, "Write M_1 to O" stands for writing the record in M_1 to the output buffer, and "Write O to Disk" stands for writing the record in O to disk. The first six operations are shown as a reference:

Read r_1 to M_1
Read r_2 to M_2
Read r_3 to M_3
Read r_4 to M_4
Write M_1 to O
Write O to Disk

(c) **[4 marks]** What is the number of block transfers and disk seeks associated with the process described in (b)?