1: Query Processing

Consider the natural join of relations Student(Name, Project) and Prof(Name, Project) on attribute Project, i.e., Student.Project = Prof.Project. The size of relation Student is 2,000 blocks and the size of relation Prof is 1,000 blocks. Assume these relations are not sorted by any of their attributes.

1. Assume that we have 201 blocks available in memory buffer, i.e., M=201. Check whether it is possible to join these relations using the sort-merge join algorithm and compute the cost of this join. Repeat the aforementioned steps for the optimized sort-merge algorithm.

2. Describe a situation where the aforementioned sort-merge join algorithms may need additional disk I/O accesses.

2: Query Processing

Consider the following relational schema.

Student (StudentID, Name)
HealthInsurance (InsuranceID, DateStarted, StudentID)

Assume that number of blocks in relation Student is 10,000 and the number of blocks in relation HealthInsurance is 300. We do not have any indexes on these relations. We like to join these relations on StudentID.

1. If we can have 12,000 blocks in main memory, i.e., M = 12,000, explain what is the fastest join algorithm to join Student and HealthInsurance on StudentID and analyze its cost?

2. If we can fit 20 blocks in main memory, i.e., M = 20, what is the fastest join algorithm to join these relations? Analyze its cost.

3: Query Processing

Assume that we like to sort a relation R. Answer the following questions.

1. If R is too large to fit in the main memory and does not have any index, explain which algorithm we can use to sort R and analyze it cost for the sort.

2. Assuming we can fit 200 blocks in main memory, i.e., M = 200, what should be the size of relation R (in blocks) to use two-pass multi-way merge-sort algorithm to sort it? What will be the maximum cost for this sort?

4: Query Optimization

Consider the following relational schema and SQL query:
Suppliers(sid, sname, address, category)
Supply(sid, pid)
Parts(pid, pname, brand)

SELECT S.sname, P.pname
FROM Suppliers S, Parts P, Supply Y
WHERE S.sid = Y.sid AND Y.pid = P.pid

1. Give two examples of join orderings that System R will not consider.

2. Enumerate all joins orderings that System R optimizer considers.

5: Query Optimization

Consider relations R(A,B) and S(B,C). Assume that R contains 2,000 tuples, and that S contains 5,000 tuples. We want to compute the equi-join \( R \bowtie S \) over attribute B.

1. Without any further assumptions, what is the maximum number of tuples that the resulting relation may contain?

2. Now assume that we know that the number of distinct values of B in R is 500. What is now a reasonable estimate on the size of join?

3. Finally, assume we know that B is a primary key in S. What is now a reasonable estimate on the size of join?