Question 1. Query Optimization and Join Algorithms

a) **[6 points]** Hybrid hash is likely to be substantially better than sort-merge for a join in which one of the relations is much larger than the other. Why is this?

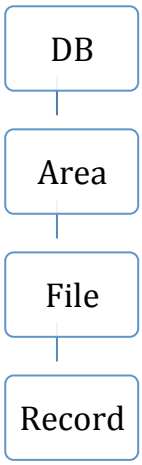
b) **[4 points]** What is an outerjoin query operation? Why is it interesting for query optimization?

c) **[4 points]** In the System R optimizer (Selinger'79) how does the number of access plans that are considered vary with the number of different join methods?

d) **[6 points]** Illustrate **one** example of a query plan that
   i. The System R optimizer (Selinger'79) will not consider in its exploration of the plan space, and
   ii. Is a plan that is more efficient that the plan space that the System R optimizer will consider.

Question 2. Concurrency Control

a) Consider the resource hierarchy shown on the right for the Gray et al.'76 scheme. Recall that in that scheme, there are six lock modes: NL, S, SIX, X, IS and IX. Now consider a transaction T1 that wants to read all the records and update a very small fraction of the records.



[2 points] What locks and in what modes would be acquired by T1 if it were to run in degree 3?

[2 point] When does T1 release the locks?

[3 points] If T1 were to run as a degree 2 (Read Committed) transaction, what locks and in what modes would it acquire?

[3 points] If T1 were to run as a degree 2 (Read Committed) transaction, when would it release each lock?

b) **[15 points]** Suppose that a DBMS recognizes increment and decrement as basic actions, in addition to reads and writes. A transaction can increment/decrement an attribute without actually knowing the actual value; i.e. increment/decrements are versions of blind writes (writes without a prior read). In addition to shared (**S**) and exclusive (**X**) locks two special locks are supported: an object must be locked in **I** mode before incrementing it, and in **D** mode before decrementing it. An **I** lock is compatible with another **I** or **D** lock on the same object, but not with **S** and **X** locks. A **D** lock is compatible with another **I** or **D** lock on the same object, but not with **S** and **X** locks.

Illustrate how the use of I and D locks can increase concurrency. Show a schedule allowed by Strict 2PL that only uses S and X locks, and explain how the use of I and D locks can allow more actions to be interleaved, while continuing to follow Strict 2PL.

c) **[7 points]** Consider the B-link protocol. What is the maximum number of latches that can be acquired by a transaction? Illustrate the scenario where this maximum number of latches is acquired.

d) **[8 points]** Consider the optimistic CC protocols proposed by Kung and Robinson. What is the CC-related overhead that read-only transactions experience in this protocol? Explain their overhead clearly in each of the 3 phases of the protocol. Note for these transaction, the transactions don't know that they will incur no writes at the start of the transaction.