Question 1. Consider the relations $R(A, B)$, $S(B, C, D)$ and $T(D, E, F)$. Translate the following SQL query to the relational algebra:

```sql
SELECT R.A, S.C FROM R, S, T
WHERE R.B = S.B AND S.D = T.D
AND S.C > 10
AND T.F NOT IN
(SELECT T2.F FROM R R2, T T2
WHERE NOT T2.E = 4
OR (T2.D = R2.A AND R2.B <= ALL
(SELECT MIN(R3.B)
FROM R R3
WHERE R3.A = R.A)
)
)
```

Use the algorithm studied in the course for this purpose. Give sub-results and motivate your answer.

Question 2. Consider the following relational algebra expression over the relations $R(A, B)$ and $S(B, C)$:

$$
(p_{R_1}(R) \times p_{R_2}(R) \times p_{R_3}(R) \times p_{S_1}(S) \times p_{S_2}(S))
$$

Optimize this expression by removing redundant joins. Give sub-results and motivate your answer.

Question 3. Explain and illustrate, by means of an example, how grid files work (how are they constructed, how can one query for points in the index, how can one do nearest-neighbour searches, how can one insert new tuples, how can one delete tuples). Be sure to explain the strengths and the weaknesses of this index.
Question 4. Consider the (clustered) relations $R(A,B,C)$, $S(C,D,E)$, $T(E,F)$ and $U(E,F)$. Records from $R$ and $S$ comprise 50, while records from $T$ and $U$ comprise 40 bytes. Blocks are 4000 bytes large and there are 11 main memory buffers available. The statistics show that $R$ contains 36000 tuples; that $S$ contains 10000 tuples; that $T$ contains 1000 tuples; and that $U$ contains 25000 tuples. Furthermore, $R.B$ is uniformly distributed in the range $[0, 100]$; $S.D$ is uniformly distributed in the range $[10000, 30000]$, and $S$ has 5000 distinct values for $E$. Relation $R$ has unclustered B-tree indexes on attributes $B$ and $C$ (separately); relation $S$ has clustered hashing indexes on attributes $D$ and $E$ (separately). Attribute $A$ is a key for $R$; $C$ is a key for $S$, and $E$ is a key for $T$ and $U$.

The query compiler has already obtained the following logical query plan:

$$\pi_{R.A, S.C, T.F} (\sigma_{R.B=50}(R) \bowtie \sigma_{S.D \geq 20000}(S) \bowtie (T \cap U))$$

Construct a sufficiently optimal physical query plan. Use disk I/Os as your optimization metric. Motivate your answer, and describe any assumptions that you make. It suffices to make only locally-optimal decisions (in other words: you may use the greedy algorithm, and your solution need not be globally optimal.)

Question 5. • Define what a conflict-serializable schedule is.

• Prove that the two-phase locking protocol is correct. That is: prove that if in a consistent and legal schedule, every transaction makes all of its lock request before any of its unlock requests, then the schedule is conflict-serializable.