Remarks:

- This test is out of 100 points. The value of each question/sub-question is written in square brackets, next to question number. You can get partial credits for your answers.
- Answer each question on a separate paper. Use only one side of each page.
- Write your name on every paper you return.


Question 2 [25] Assume that we have two relations \( R \) and \( S \) with sizes \( B(R) \leq B(S) \). Assume that we have a main memory of size \( M \) which is less than \( B(R) \).

A [5] Compute the number of disk I/Os needed for block nested loop join (i.e., BNLJ).

B [10] A variation of the BNLJ, named balanced BNLJ (or BBNLJ), allocates half of the main memory for \( R \) and the remaining half for \( S \) for the join operation. Compute the number of disk I/Os needed for BBNLJ.

C [10] Dr. Genius thinks that BBNLJ can be preferable over BNLJ if the data is clustered on disk. Dr. Formula thinks that BNLJ is always the better of the two algorithms. Who do you agree with, and why?

Question 3 [20] In this question, you will assume that the size of each of the relations is larger than the available memory, and thus, two pass algorithms are required.

Hybrid hash join is an improvement over standard hash join. The basic idea behind hybrid hash join is that it keeps one or more hash lists in main memory when it makes the first pass over the the two relations that are being joined. As a result, it reduces the number of disk I/Os for those hash lists from the two relations.

Can we adapt the hybrid hash join idea to improve the following algorithms? If your answer is “yes”, briefly describe how the new hybrid algorithm works. Also discuss how much disk I/O is saved. If your answer is “no”, briefly explain why it is not possible.
A [10] Hash-based set union \( R \cup set S \).

B [10] Sort-based join \( R \bowtie S \).

**Question 4** [25] Answer the questions below for the following schedule:

\[ S = w_1(C); w_2(A); r_3(B); r_2(B); r_1(A); r_3(A); w_3(B); r_1(B); \]


C [6] Show how \( S \) will behave if 2PL is used.

D [6] Show how \( S \) will behave if shared and exclusive locks are used.

E [6] Show how \( S \) will behave if upgrade locks are used.

**Question 5** [15] Assume that time stamps are used for concurrency. Assume that there are three transactions \( T_1, T_2 \) and \( T_3 \) with start times \( TS(T_1) = 10, TS(T_2) = 20 \) and \( TS(T_3) = 30 \). The table below shows the order in which the actions of these transactions take place. Write what happens after each of the actions. Also, write the read and write times (RT and WT) of all the variables whenever they change. You will assume that the read and write times of all the variables are initially zero. You can use this sheet to answer this question.

<table>
<thead>
<tr>
<th>( T_1 )</th>
<th>( T_2 )</th>
<th>( T_3 )</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_1(C) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_2(C) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( r_2(B) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_3(B) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_2(A) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w_1(B) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( r_3(A) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( r_1(A) )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>