Warmup #1  (Textbook Problem 13.15)

Suppose that a B+-tree index on \( (\text{dept name}, \text{building}) \) is available on relation \( \text{department} \). What would be the best way to handle the following selection?

\[
\sigma(\text{building}<\text{"Watson"}) \land (\text{budget}<55000) \land (\text{dept name}=\text{"Music"}) (\text{department}).
\]

Warmup #2  (Textbook Problem 14.7)

What is a cascadeless schedule? Why is cascadelessness of schedules desirable? Are there any circumstances under which it would be desirable to allow noncascadeless schedules? Explain your answer.

Warmup #3  (Textbook Problem 14.12)

List the ACID properties. Explain the usefulness of each.

Warmup #4  (Textbook Problem 14.15)

Consider the following two transactions:

\[
\begin{align*}
T_{13} & : \quad \text{read}(A); \\
& \text{read}(B); \\
& \quad \text{if } A = 0 \text{ then } B := B + 1; \\
& \text{write}(B). \\
T_{14} & : \quad \text{read}(B); \\
& \text{read}(A); \\
& \quad \text{if } B = 0 \text{ then } A := A + 1; \\
& \text{write}(A).
\end{align*}
\]

Let the consistency requirement be \( A = 0 \lor B = 0 \), with \( A = B = 0 \) the initial values.

1. Show that every serial execution involving these two transactions preserves the consistency of the database.

2. Show a concurrent execution of \( T_{13} \) and \( T_{14} \) that produces a nonserializable schedule.

3. Is there a concurrent execution of \( T_{13} \) and \( T_{14} \) that produces a serializable schedule?
Problem 5 (To Be Graded)

A database stores warehouses and products. Here is the schema:

```sql
create table Warehouse(
    name varchar(100) primary key,
    zip_code varchar(20) not null,
    index(zip_code));

create table Product(
    id int primary key,
    description varchar(600) not null,
    availableAt varchar(100),
    index(availableAt),
    foreign key(availableAt) references Warehouse(name)
);
```

There are 500,000 products and 10,000 warehouses in the database. The average size of a warehouse record is 50 bytes and the average size of a product record is 100 bytes. All indexes are B+-trees. There are 500 known zip codes. You want to find all products stored in Providence warehouses at the zip code "02904-2413".

Consider the following query plans:

1. Select Providence warehouses (02904-2413), and then join with product using an index nested loop join.
2. Select Providence warehouses (02904-2413), and then join with Product using a sort-merge join.
3. Join Product with Warehouse using an index nested loop and then select the merchandise in (02904-2413).

With the assumptions that:

- An index lookup takes approximately 20ms.
- Scanning through a table takes 0.1 ms per block.
- The disk block size is 8K = 8192 bytes.
- Buffer size is 4 blocks.

For each query plan, find the approximate runtime; discuss which plan should be selected and why.
Problem 6 (To Be Graded)

Consider the following transactions. Operations are to be executed in order from top to bottom.

<table>
<thead>
<tr>
<th>Schedule A</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>write(H)</td>
</tr>
<tr>
<td>read(G)</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>write(E)</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>-</td>
</tr>
</tbody>
</table>

Is this schedule conflict serializable? Prove your answer by building a precedence graph. If serializable, provide a possible serialized order of transactions.

**NOTE:** Your graph must be large enough (and neat enough) to be read in order to receive credit for this problem. Please provide labels indicating the dependency(s) that each arrow represents.