1. Create a schema in XML Schema corresponding to the following DTD:

```xml
<!DOCTYPE bib [
  <!ELEMENT book (title, author+, publisher, keyword+)>
  <!ELEMENT publisher (pub-name, pub-branch) >
  <!ELEMENT title ( #PCDATA )>
  <!ELEMENT author ( #PCDATA )>
  <!ELEMENT keyword ( #PCDATA )>
  <!ELEMENT pub-name( #PCDATA )>
  <!ELEMENT pub-branch( #PCDATA )>
]>
```

2. Given relations r(A, B, C) and s(C, D, E), which have the following properties: r has 40,000 tuples, s has 30,000 tuples, 50 tuples of r fit on one block, and 15 tuples of s fit on one block. Estimate the number of block transfers and seeks required, using each of the following join strategies for r ⋈ s:
   a. Nested-loop join
   b. Block nested-loop join
   c. Indexed nested-loop join (suppose there is a primary B+ tree index with height 4 on the join attribute (C).)
   d. Merge join
   e. Hash join

3. Suppose that a B+-tree index on (branch-name, branch-city) is available on relation branch. What would be the best way to handle the following selection?

\[ \sigma(\text{branch-city} < \text{“Brooklyn”}) \land (\text{assets} < 5000) \land (\text{branch-name} = \text{“Downtown”}) (\text{branch}) \]

4. Exercise 15.6. (15.10 in old version). Also, suggest to either add a new edge (i.e., <T1,T3>) or remove an edge that would change the story (reverse your conclusion of conflict serializable or not).

5. Consider the following schedule:

<table>
<thead>
<tr>
<th>Step</th>
<th>T₀</th>
<th>T₁</th>
<th>T₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Write(A)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Write(A)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>Write(A)</td>
</tr>
<tr>
<td>4</td>
<td>Write(B)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Write(B)</td>
<td></td>
</tr>
</tbody>
</table>

Is this schedule allowed in timestamp-ordering protocol? Explain why?