CS 3743
Introduction to Database Systems
Midterm 3

<table>
<thead>
<tr>
<th>Question</th>
<th>Mark</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NAME:__________________________

Instructions
1. Do all of the 5 problems
2. You have 60 minutes for the exam
4. Show all your work
5. Do not separate exam papers

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy</td>
<td>Difficulty Level</td>
<td>Difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. *Concurrency Control*: (20 pts, 6+7+7 pts)

(a) In below schedule \(l(x)\) denotes lock on item \(x\) and \(u(x)\) denotes unlock on item \(x\). \(c_i\) denote commit of transaction \(i\). Does the following schedule obey two-phase locking? Explain briefly.

\[
\begin{align*}
T1: & \quad l(x) \quad r(x) \quad u(x) \\
T2: & \quad l(x) \quad l(y) \quad r(x) \quad w(x) \quad r(y) \quad u(y) \quad u(x) \quad c_2
\end{align*}
\]

(b) Consider the following two transactions. Can execution of these transactions using two-phase locking lead to deadlock? Explain briefly.

\[
\begin{align*}
T1: & \quad r(x); \quad r(y) \\
T2: & \quad r(x); \quad w(x); \quad r(y); \quad w(y)
\end{align*}
\]

(c) Can deadlock occur of timestamp-based concurrency control is used? Explain briefly.
2. *Disk Storage*: (20 pts, 4 pts each) A file has \( r = 20000 \) student records of fixed length. Each record has the following fields: name (30 bytes), ssn (9 bytes), address (40 bytes), phone (9 bytes), birthdate (8 bytes), sex (1 byte) and department code (4 bytes). An additional byte is used as a deletion marker. The file is stored on a disk with following parameters. \( B = 512 \) bytes, 20 blocks per track, 400 tracks per surface. Disk pack consists of 15 double-sided disks, drive rotates at 2400 rpm (revolutions per minute) and average seek time of 30 msec.

Answer the following questions based on above information

(a) What is the record size \( R \) in bytes?

(b) What is the blocking factor \( bfr \)?

(c) What is the number of file blocks assuming an unspanned organization?

(d) What is the average time to find a record by doing linear search on the file if file blocks are stored contiguously.

(e) Assume that the file is ordered by ssn. What is the time it takes to search for a record given its ssn value by doing a binary search.
3. **Transaction Processing (20 pts):** Consider the three transactions $T_1$, $T_2$ and $T_3$ and the schedule $S$ given below.

$T_1 : r_1(A); r_1(D); w_1(A); c_1$

$T_2 : r_2(D); r_2(B); w_2(D); w_2(B); c_2$

$T_3 : r_3(A); r_3(B); w_3(B); c_3$

$S : r_1(A); r_3(A); r_1(D); r_2(D); r_3(B); w_1(A); c_1; w_3(B); c_3; r_2(B); w_2(D); w_2(B); c_2$

(a) Draw the serializability (precedence) graph for $S$. Is $S$ is serializable?

(b) Is $S$ strict?

(c) Is $S$ cascadeless?

(d) Is $S$ recoverable?
4. **Indexing Structures:** (20 pts) Given a point \( p \), nearestneighbor query \( nn(p) \) returns the item in the database that is closest to \( p \) according to some metric such as absolute value. For example, consider the set \( \{1, 4, 7\} \). The query \( nn(6) \) will return 7 since \( |7 - 6| = 1, |4 - 6| = 2 \) and \( |1 - 6| = 5 \) and 1 is the smallest distance. Your are given 1000 records. Each record is 100 bytes long. Each record includes a point plus some additional information. Block size is 1000 bytes. All the point coordinates are distinct and are integers in \( \{1, 2, 3, ...5000\} \). You have full control over how the file is structured, stored on the disk. Assume each point is 4 bytes and block pointer is 6 bytes. Suggest an index structure that speeds up nearest neighbor queries and show how the file is organized and stored to use your index. To show the speedup, compare the number of blocks retrieved without index and number of blocks retrieved with index.
5. *Query Optimization*: (20 pts, 6+7+7pts) Consider the following scheme where the keys are underlined.

EMPLOYEE(SSN, Name, Dept)

PROJECT(SSN, PID, Name, Budget)

Consider the following query

```
Select P.budget, P.name, E.name
From Employee E, Project P
Where E.ssn = P.ssn and
    P.budget > 99 and
    E.name = 'John';
```

Assume the following information

- 10000 tuples in employee relation.
- 20000 tuples in project relation.
- 1000 different values for E.name and values are uniformly distributed.
- The domain of budget consists of integers in the range 1 to 100.
- Employee has an index on Name and another index on SSN.
- Project has an index on SSN and another index on budget.

Answer the following questions

(a) Draw the canonical query tree for above query.
(b) Draw an efficient query tree for above query.

(c) Explain how the available indexes speed up execution of this query.