Datenbanken II (SS 2017)
Exercise 10

Task 1 Discuss advantages and disadvantages of randomized approaches for join-order optimization compared to deterministic approaches. Why are randomized approaches need in practice?

Task 2 Which problem does the iterative Improvement have? How can this problem be solved?

Task 3 How can the join order be encoded within genetic algorithms? Which properties do mutation and crossover operations need to fulfill with respect to this encoding?

Task 4 Bitmap-Index
(a) Explain the functionality of a Bitmap-Index.
(b) Create a Bitmap-Index for the following table (discuss different possibilities). Which queries are you able to support with it?

<table>
<thead>
<tr>
<th>Name</th>
<th>MatrNr</th>
<th>Sex</th>
<th>Start of Study</th>
<th>Study Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tom</td>
<td>15874</td>
<td>M</td>
<td>2002</td>
<td>INF</td>
</tr>
<tr>
<td>Claudia</td>
<td>18091</td>
<td>W</td>
<td>2006</td>
<td>CV</td>
</tr>
<tr>
<td>Klaus</td>
<td>14342</td>
<td>M</td>
<td>2003</td>
<td>INF</td>
</tr>
<tr>
<td>Susanne</td>
<td>11012</td>
<td>W</td>
<td>2001</td>
<td>INF</td>
</tr>
</tbody>
</table>

(c) How does a Bitmap-Index support exact-match, partial-match and range queries?
(d) How many Bitvectors do you need to encode the Immatriculation numbers?

Task 5 KdB/kd-Baum
(a) Insert the following data series into a kd-tree:
   - Data 1: (10 Z), (30 A), (20 Z), (20 A), (30 Z), (10 A), (99 H)
   - Data 2: (10 A), (15 B), (30 C), (45 H), (50 I), (55 K), (60 O), (90 T)
   - Data 3: (10 Z), (20 K), (30 A), (40 L), (50 M), (60 N), (70 D), (80 X)
(b) Draw the corresponding brickwall.
(c) What is a KdB-Tree?

Good Luck!