National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Advance Database Concepts**	**Course Code:**	**CS 4064**
Program:	**BS (SE)**	**Semester:**	**Spring 2025**
Due Date:	**7th of April, 2025.**	**Total Marks:**	**100**
Section:	**6A, 8A**	**Page(s):**	**4**
Type:	**Assignment 2**		

Important Instructions:

Each student must solve the assignment using their unique last four digits of their roll number (denoted as **XXXX**).Show all steps and formulas clearly. Submit a HARD COPY of your assignment, submission in soft copy is not required.

Question 1: Disk Storage & File Organization

A university maintains a **STUDENT** file with **r = (5000 + XXXX) records** of fixed-length. Each record has the following fields:

- **NAME (30 bytes), ROLLNO (9 bytes), ADDRESS (40 bytes), PHONE (9 bytes), BIRTHDATE (8 bytes), SEX (1 byte), PROGRAMCODE (4 bytes), MAJORCODE (4 bytes), YEARCODE (4 bytes), and DEGREE (3 bytes).**
- An additional **1-byte deletion marker** is included.
- The file is stored on a disk with the following parameters: **Block size (B) = 512 bytes, Seek time = 20 ms**, **Rotational delay = 10 ms**, **Block transfer time = 1 ms**

Tasks:

1. **Calculate the record size (R) in bytes.**
2. **Determine the blocking factor (bfr) and the number of file blocks (b) assuming an unspanned organization.**
3. **Calculate the average search time for a record:**
    - **(i) If blocks are stored contiguously (double buffering used)**
    - **(ii) If blocks are randomly distributed**
4. **If the file is sorted by ROLLNO, estimate the binary search time to find a record.**

Question 2: Indexing Techniques

A company stores **(20000 + XXXX) EMPLOYEE records** in a file with:

- **Fixed-length records**
- **Each record size = 115 bytes**
- **Block size B = 512 bytes**
- **SSN (9 bytes) is the key field**
- **Pointer size P = 6 bytes**
- **Record pointer P_r = 7 bytes**

Tasks:

1) **Compute the blocking factor (bfr) and the number of file blocks (b).**
2) **Assuming the file is ordered by SSN, construct a primary index:**
    a) Calculate the **index blocking factor (bfr_i)**.
    b) Determine the **number of first-level index entries and blocks**.
    c) Compute the **number of levels in a multi-level index**.
    d) Find the **total blocks required by the multi-level index**.
    e) Compute **block accesses needed to search for an SSN using this primary index**.
3) **Repeat the above calculations for a secondary index (if the file is unordered). Compare with the primary index.**
4) Suppose the file is ordered by the non-key field Department_code and wewant to construct **a clustering index on PROGRAMCODE** that uses blockanchors (every new value of PROGRAMCODE starts at the beginning of a newblock). Assume there are 1000 distinct values of PROGRAMCODE, and that the STUDENT records are evenly distributed among these values. **Part 'e' will be modified as:**
    a) the number of block accesses needed to search for and retrieve all records in the file having a specific PROGRAMCODE value using the clustering index (assume that multiple blocks in a cluster are either contiguous or linked by pointers).

Question 3: Static Hashing

A **PARTS** file contains **certain number of records, hashed using the following function:**

$$h(K)=K \bmod 8$$

Each **bucket is one disk block** and holds **two records**.

The following **Part# values** need to be hashed into the file:

2369,3760,4692,4871,5659,1821,1074,7115,1620,2428,3943,4750,6975,4981,9208

Tasks:

1. **Load the records into the hash file using h(K)=K mod 8.**
    - Assign each record to the appropriate bucket.
    - Show which records go into overflow.
2. **Determine the number of overflow records.**
3. **Calculate the average block accesses required for a random retrieval.**

Question 4: B+ Tree Indexing

A file contains **N records** with each record having an **SSN (Social Security Number) as the key field**. The file is **not ordered by SSN**, and we want to construct a **B+-tree index** on the SSN field.

Each **student must use the last four digits of their roll number (XXXX) to personalize the problem**:

- **Number of records (N) = 50,000 + XXXX**
- **Block size (B) = 4 KB (4096 bytes)**
- **SSN key size = 9 bytes**
- **Pointer size = 6 bytes**
- **Record size = 120 bytes**

Using this data, answer the following:

1. **Compute the orders of the B+-tree:**
    - **Internal node order (p):** Maximum number of **key-pointer pairs** that fit in one **4 KB block**.
    - **Leaf node order (p_leaf):** Maximum number of **records per leaf node** that fit in a **4 KB block**.
2. **Calculate the number of leaf-level blocks** required, assuming each leaf block is **49% full** (round up).
3. **Determine the number of B+-tree levels** required if **internal nodes are also 49% full** (round up).
4. **Compute the total number of blocks** (internal + leaf) required for the B+-tree.
5. **Estimate the number of block accesses** needed to **search for and retrieve** a record given its **SSN value** using the B+-tree index.

Question 5: B Tree Indexing

Repeat Question 4, but for a B-tree rather than for a B+-tree. Compare your results for the B-tree and for the B+-tree.

Question 6: Extendible and Dynamic Hashing

Show the working for hashing the following values using:

237, 589, 124, 763, 901, 456, 378, 812, 645, 293, 715, 847, 362, 508, 194

- Hash Function value%8 for Extendible Hashing where each bucket can store up to 3 records.
- Hash Function (value+10)%3 for Dynamic hashing where each bucket can store up to 3 records.

Question 6: B+ Tree formation

A PARTS file with Part# as the key field includes records with the following Part# values:
23, 65, 37, 60, 46, 92, 48, 71, 56, 59, 18, 21, 10, 74, 78, 15, 16, 20, 24, 28, 39, 43, 47, 50, 69, 75, 8, 49, 33, 38.

Suppose that the search field values are inserted in the given order in a **B+-tree of order p=4 and pleaf=3**; show how the tree will expand and what the final tree will look like.

----------*Best of Luck*----------