CS54100 - Homework 2
FALL 2016

Due: Monday, October 31, 2016 in class, before class starts.
(There will be a 10% penalty for each late day. After 5 late days, the homework will not be accepted.)

Part 1: Tree-Based Indexing (30 Points)

1. Show the result of inserting 15, 37, 24, 36, 32, 11, 28, 39, 18, 27, 38 and 2 into an initially empty B+-tree of order 3. Show all steps of insertion (you do not need to show a separate tree if insertion does not cause any splits).

2. Show the result of deleting 32, 27, 11 from the previous B+-tree. Show the new B+-tree after each deletion, even if no redistributions or merges happen.

3. Assume that you have just built a dense B+ tree index using Alternative (2) on a heap file containing 30,000 records. The key field for this B+ tree index is a 40-byte string, and it is a candidate key. Pointers (i.e., record ids and page ids) are (at most) 10-byte values. The size of one disk page is 1000 bytes. The index was built in a bottom-up fashion using the bulk-loading algorithm, and the nodes at each level were filled up as much as possible. Answer the following:

    a. How many levels does the resulting tree have? Show your steps.

    b. For each level of the tree, how many nodes are at that level? Show your steps.

    c. How many levels would the resulting tree have if key compression is used and it reduces the average size of each key in an entry to 10 bytes? Show your steps.

4. If your database system supported both a static and a dynamic tree index (e.g, ISAM and B+ trees), would you ever consider using the static index in preference to the dynamic index? why?

Part 2: Hash-based Indexing (25 Points)

Give examples of the following by showing the insertions step-by-step:

1. A Linear Hashing index and an Extendible Hashing index with the same data entries, such that the Linear Hashing index has more pages.

2. A Linear Hashing index and an Extendible Hashing index with the same data entries, such that the Extendible Hashing index has more pages.

State clearly any assumptions you may assume.

Part 3: Query Evaluation (20 Points)

Consider the following schema with the Employees relation:
Employees(EmpId: Integer, EmpName: String, Salary: Real)

For each of the following indexes, list whether the index matches the given selection conditions. If there is a match, list the primary conjuncts. Please state clearly any assumptions you may assume.

1. A hash index on the search key (Employees.EmpId).
   (a) $\sigma_{Employees.EmpId = 30,000}(Employees)$
   (b) $\sigma_{Employees.EmpId > 10,000}(Employees)$

2. A B+-tree index on the search key (Employees.EmpId).
   (a) $\sigma_{Employees.EmpId = 20,000}(Employees)$
   (b) $\sigma_{Employees.EmpId > 20,000}(Employees)$

3. A hash index on the search key (Employees.EmpId, Employees.Salary).
   (a) $\sigma_{Employees.EmpId < 15,000 \text{ And } Employees.Salary > 5000}(Employees)$
   (b) $\sigma_{Employees.EmpId = 6000 \text{ And } Employees.Salary = 2500}(Employees)$
   (c) $\sigma_{Employees.EmpId = 20,000}(Employees)$
   (d) $\sigma_{Employees.Salary = 6000}(Employees)$

4. A B+-tree index on the search key (Employees.EmpId, Employees.Salary).
   (a) $\sigma_{Employees.EmpId > 45,000 \text{ And } Employees.Salary = 2000}(Employees)$
   (b) $\sigma_{Employees.EmpId = 4,000 \text{ And } Employees.Salary > 9000}(Employees)$
   (c) $\sigma_{Employees.EmpId = 8}(Employees)$
   (d) $\sigma_{Employees.Salary = 5500}(Employees)$