CSE 4701 Spring 2020

Homework 2: Due March 4 (Wed) 2020, midnight, at HuskyCT

Problem 1. (35 pts)

Consider a disk with block size B = 512 bytes. A block pointer is P = 6 bytes long, and a record pointer is P_R = 7 bytes long. A file has r = 100,000 EMPLOYEE records of fixed length. Each record has the following fields: NAME (25 bytes), SSN (9 bytes), DEPARTMENTCODE (9 bytes), ADDRESS (40 bytes), PHONE (9 bytes), BIRTHDATE (8 bytes), SEX (1 byte), JOBCODE (5 bytes), SALARY (4 bytes).

a. Calculate the record size R in bytes.

b. Calculate the blocking factor *bfr* and the number of file blocks *b*, assuming an unspanned organization.

c. Suppose that the file is ordered by the key field SSN and we want to construct a primary index on SSN. Calculate

(i) the index blocking factor *bfri*

(ii) the number of first-level index entries and the number of first-level index blocks

(iii) the number of levels needed if we make it into a multilevel index

(iv) the total number of blocks required by the multilevel index, and

(v) the number of block accesses needed to search for and retrieve a record from the file—given its SSN value—using the primary index.

Problem 2. (15 pts)

Given the same specifications of Problem 1, consider this time you are building a primary index on SSN using B-tree. Calculate (i) the order *p* for the B-tree, (ii) the number of levels needed if blocks are approximately 69% full (round up for convenience), and (iii) the worst-case number of blocks needed to search for and retrieve a record from the file—given its SSN value—using the B-tree you are estimating.

Problem 3. (25 pts)

A PARTS file with Part# as key field includes records with the following Part# values: 12, 33, 22, 15, 19, 55, 25, 11, 16, 13, 29, 44, 71, 21, 4, 42, 68, 78. Suppose that the search field values are inserted in the given order in a B+-tree of order p = 4 and p_{leaf} = 3.
(i) Show how the tree will expand (show all steps as in Fig 18.12 (6^{th} ed)) and what the final tree will look like.
(ii) What is the fill ratio of the final B+-tree you created? (Note: we learned 69% is the average fill ratio in class.) Here the fill ratio is defined as (# of filled key values)/(# of maximally allowed key values) regardless each key value resides in index node or sequence set node.

Note: 5% penalty for one day late submission