

SOFTENG 351 S1 C : Assignment 3

Due Date: Sunday 24 May 2020 at 4:59pm

50 marks in total = 5% of the final grade

1. Consider a disk with the block size of 512 bytes. A block pointer takes 8 bytes while an integer takes 4 bytes. A file has 30,000 EMPLOYEE records of fixed length. Each record has the following fields: Name, Ssn, DepartmentCode, Address, Phone, BirthDate, Sex, JobCode and Salary. The size of each record is 115 bytes.
 - (a) Calculate the blocking factor of the data file and the number of blocks in the data file. [2 marks]
 - (b) Suppose that the file is ordered by the key field Ssn which is an integer. A primary index on Ssn is constructed. Calculate
 - i. The size of each index entry. [1 marks]
 - ii. The blocking factor of the index file. [2 marks]
 - iii. The number of first-level index entries and the number of first-level index blocks. [2 marks]

2. Suppose that one block can hold either ten records or 99 keys and 100 pointers. We can use B^+ -trees as part of several different structures. For each structure described below, determine

- (a) The total number of blocks needed for a 1,000,000-record file, and
- (b) The average number of disk I/Os to retrieve a record given the search key.

You may assume nothing is in memory initially, and the search key is the primary key for the records.

- (a) The data file is a sequential file, sorted on the search key, with 10 records per block. The B^+ -tree is a sparse index where each internal node is full (99 keys and 100 pointers). [3 marks]
- (b) The same as (a), but the data file consists of records in no particular order, packed 10 to a block. The B^+ -tree is a dense index. [4 marks]
- (c) Assume that the average B^+ -tree node is 70% full; i.e., it will have 69 keys and 70 pointers while the data node has 7 records. [4 marks]

3. Insert into an initially empty B^+ -tree with 6 records with the following key values **sequentially**:

23, 26, 43, 77, 4, 92.

Each block can hold up to two data records. Each block can hold up to 3 pointers and 3 integers, i.e., the blocking factor of the index file is 3. Draw the snapshots of the B^+ -tree before and after the insertion of 43, 4, 92, respectively. [16 marks]

4. Suppose that we are using extendible hashing to insert, to an originally empty file, records with the following key values sequentially:

50, 44, 25, 20, 37, 7, 51, 49, 18, 69.

Let the hash function $h(x)$ be the **3-digit** binary code of $(x \bmod 7)$ which automatically complements the code with 0 from the left side if the code has less than 3 digits. For example, $h(1)$ is 001. Assume that each block can hold up to three records. In constructing the hashing table, the binary bits of $h(x)$ is used from left to right. Show the extendible hashing structure after the insertion of 37, 7, 69, respectively. [16 marks]