CMPUT 291 - File and Database Management (Fall 2019)

Exercises on disks and indexing

A few exercises related to disk and indexing

Q1. Consider an empty B+-tree where leaves can hold up to 2 entries and non-leaves can hold up to 3 entries.

a) Insert into your index 12 records with keys chosen as random numbers in 1 to 200.
b) Select 4 records randomly from the index and delete them.

Q2. Consider an empty extendable hash file where each bucket can hold up to 3 entries.

a) Insert 12 records with keys chosen as random numbers in 1 to 200.
b) select 4 records randomly from the index and delete them.

Q3. Consider a disk with 930,408 cylinders, 6 tracks per cylinder, 128 sectors per track and 512 bytes per sector. What is the disk capacity in bytes?

Q4. Consider a disk with 128 sectors per track, 512 bytes per sector and average seek time of 8 msec. The disk platters rotate at 5400 rpm. How long does it take to read a block of 20 sectors?

Q5. Consider a B+-tree index where each node (leaf and non-leaf) can store up to 200 entries, and you want to insert 10 million index entries to the index.

a) Find the minimum and maximum depth of the tree?
b) Suppose the index is built from scratch using the insert algorithm discussed in class; without worrying about the effects of buffering, estimate the number of I/Os in the worst case.
c) Do the estimation in part (b) but assume the first two levels of the index are stored in main memory.
d) Suppose you can freely sort the index entries! What would be your best algorithm to load the index? Again estimate the number of I/Os without worrying about the effects of buffering.

Q6. Consider the following query

*select * from weather where city = "Edmonton" and temp < -30*

Suppose B+-tree indexes are constructed on columns city and temp. Data entries are stored in both indexes as (key,rid,pid) where rid is the record id of a data record with key value k and pid is the address of a page where the record with id rid is stored. Further assume 1% of the tuples satisfy the predicate "city='Edmonton'", 1% of the tuples satisfy the predicate "temp<-30" and 0.01% of the tuples satisfy both predicates.

a) What would be an efficient algorithm to evaluate the query if we know that the index on city is clustered?
b) What would be an efficient algorithm to evaluate the query if we know that none of the indexes are clustered?
c) Suppose each page of the index (leaf and non-leaf) stores up to 200 keys, and each data page stores up to 100 records. Assume both index and data pages are full. If N denotes the number of data records, estimate the number of page accesses for both algorithms given in (a) and (b).

Last modified: Friday, 1 December 2017, 11:26 AM

◄ Mini-project 2 demo booking

Jump to...