CAS CS 560: Introduction to Database Systems
Homework #3
Spring 2003

Due Date and Time: March 28, 2003 at 4:00 PM

Read Chapters 11 and 12 from the textbook and solve the following problems:

Problem 1. [20 pts]
Consider a disk with sector size of 512 bytes, 10,000 tracks per surface, 800 sectors per track, 5 double-sided platters.

1. What is the capacity of a track in bytes? What is the capacity of a surface? What is the capacity of the disk?
2. How many cylinders does the disk have?
3. If the disk platters rotate at 11,000 rpm (revolutions per minute), what is the maximum and average rotation latency?
4. Assuming that one track of data can be transferred per revolution, what is the transfer rate?

Problem 2. [25 pts]
Construct a B+-tree for the following set of key values:

(1, 3, 2, 8, 6, 10, 9, 7, 5, 11)

a) Assume that the tree is initially empty and values are added in the order given above. Construct B+-trees for the cases where the number of pointers that will fit in one node is **four (4)** and **six (6)**.

b) Show the state of the tree after each of the following operations is applied to the tree created after the insertions:

1. Delete 6.
2. Delete 7.
3. Delete 5.

Problem 3. [30 pts]
Suppose that blocks (pages) can hold either ten records or 99 keys and 100 pointers. Also assume that the average B+-tree node is 70% full, i.e. it will have 69 keys and 70 pointers. We can use B+-trees as part of several different structures. For each structure described below, determine (i) the total number of blocks needed for a 1,000,000-record file, (ii) the average number of disk I/O's to retrieve a record given its search key, (iii) the average number of disk I/O's to retrieve records for a range query that is matched by 1000 records. You may assume that nothing is in memory initially, and the search key is the primary key for the records:

1. The data file is a sequential file, stored on the search key, with 10 records per block. The B+-tree is a dense index.
2. The same as above but the B-tree is a sparse index.
3. The same as (1), but the data file consists of records in no particular order (heap file), packed 10 to a block.
4. Instead of the B+-tree leaves having pointers to data records, the B+-tree leaves contain the records themselves. A block can hold ten records, but on the average, a leaf block is 70% full.

Problem 4. [25 pts]
Suppose that we are using extensible hashing on a file that contains records with the following search-key values:

$$(3, 2, 5, 15, 19, 17, 26, 21, 55)$$

a) Show the extensible hash structure after each pair of insertions for this file if the hash function is h(x) = x mod 8 and buckets can hold three (3) records.

b) Show the extensible hash structure after each of the following operations:

1. Delete 19.
2. Delete 15.
3. Delete 55.