BBM 371 – Data Management
Exercises

1. You have got 1000 records that are fixed length and 100 bytes. The records will be stored on a disk with a sector size of 102 bytes. Each cluster involves 4 sectors and each track involves 60 sectors. The rotational time is 4000 rpm and the average seek time is 6 msec. Answer the following questions.

a) Blocking factor of the files.

b) Number of blocks to store the file.

c) Average time to find a record if the read/write head is at a random position initially and the file is sorted (cylinder based approach is not used).

d) Time to add a new record into the file if the read/write head is at a random position initially and the file is sorted. Consider the worst case in your solution and assume that the number of blocks in the sorted file will remain the same once the new record is inserted.

e) We want to index the given file with a dense index. Assume that index stores a search key of 4 bytes and a pointer of 4 bytes. How many disk accesses do we need in order to find a record?

f) We want to index the given file with a sparse index. Assume that index stores a search key of 4 bytes and a pointer of 4 bytes. How many disk accesses do we need in order to find a record?

g) Imagine we have 5 available buffer pages and our file is a heap file. We want to sort the file. How many runs will be produced in the first pass?

h) How many passes will it take to sort the file completely?

i) What is the total IO cost of sorting the file?

2. A database uses a Linear Hashing based index. Answer the following questions if the Next pointer is pointing to bucket 25 (buckets are indexed starting from 0).

a) What is the minimum number of primary buckets in this Linear Hash?

b) If the number of primary buckets is as calculated in part (a) how many bucket splits are needed to split the 3rd bucket.

c) If Next points to 25 and the number of primary buckets is 89 which buckets are accessed for searching the following two keys k_1 and k_2, clearly show how you calculate the bucket index from the hash values indicating the bits you are using (Use the least significant bits for addressing).

$H(k_1) = 175$
$H(k_2) = 83$

3. Suppose that block size is 200 bytes, each record is 12 bytes, each key is 4 bytes, and each pointer is 4 bytes. We have got 1200 records to be indexed by using a B+ tree.

   a) How many leaf nodes are needed in order to index the given file by using a B+ tree? Remember that each leaf node stores the keys and records (not the pointers to the records).

   b) Imagine each B+ tree leaf node is %50 full. How many leaf nodes are needed in order to index the given file by using a B+ tree?

   c) How many disk accesses is needed for an equality search on a given key if only the root node and the first level of the tree are stored in the memory? Imagine each B+ tree leaf node is completely full.

   d) How many disk accesses is needed for a range search operation if the records for the range search are stored in 15 different nodes and if only the root and the first level of the tree is stored in the memory? Imagine each B+ tree leaf node is completely full.