ICS 324: Assignment # 5

1. Consider a disk with the following characteristics: block size *B* = 512 bytes; number of blocks per track = 20; number of tracks per surface = 400. A disk pack consists of 15 double-sided disks. (Assume 1 block = 1 sector)

   a. What is the total capacity of a track?

      512 * 20 = 10,240 bytes

   b. How many cylinders are there?

      400

   c. What are the total capacity of a cylinder?

      10240 * 15 * 2 = 307,200 bytes

   d. What are the total capacity of the disk?

      400 * 307200 = 122,880,000 bytes

   e. Suppose that the disk drive rotates the disk pack at a speed of 2400 rpm (revolutions per minute);

      i. what are the transfer rate (*tr*) in bytes/msec?

         tr = 2400 * 10240 /(60 * 1000) = 409.6 bytes/msec

      ii. What is the block transfer time (*btt*) in msec?

         btt = 512/409.6 = 1.25 msec

      iii. What is the average rotational delay (*rd*) in msec?

         Rd = 1.25 * 20 = 25 msec

   f. Suppose that the average seek time (st) is 30 msec. How much time does it take (on the average) in msec to locate and transfer a single block, given its block address?

      St + rd + btt = 30 + 25 + 1.25 = 56.25 msec

   g. Calculate the average time it would take to transfer 20 random blocks, and compare this with the time it would take to transfer 20 consecutive blocks. Assume a seek (st) time of 8 msec.

      20 random blocks = 20 * (8 + 25 + 1.25) = 685 mesc

      20 Consecutive blocks = 8 + 25 + 20*1.25 = 58 msec

2. A file has *r* = 20,000 STUDENT records of *fixed length.* Each record has the following fields: Name (30 bytes), Ssn (9 bytes), Address (40 bytes), PHONE (10 bytes), Birth_date (8 bytes), Sex (1 byte), Major_dept_code (4 bytes), Minor_dept_code (4 bytes), Class_code (4 bytes, integer), and Degree_program (3 bytes). The file is stored on the disk whose parameters are given in the previous question.

   a. Calculate the record size *R* in bytes.

      Record size = 30 + 9 + 40 + 10 + 8 + 1 + 4 + 4 + 3 = 109 bytes

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

      bfr = floor(512/109) = 4

   c. Calculate the average time it takes to find a record by doing a linear search on the file if

      i. the file blocks are stored contiguously

         blocks = roof(20000/4) = 5000
         average time = st + rd = 2500 *btt = 30 + 25 + 2500 * 1.25 = 3180 msce

      ii. the file blocks are not stored contiguously.

         Average time = 2500 * (30 + 25 + 1.25) = 140,625 msec

   d. Assume that the file is ordered by Ssn; by doing a binary search, calculate the time it takes to search for a record given its Ssn value.

      Random blocks accessed = $Log_2(5000)$ = 13 blocks
      Time to read 13 random blocks = 13 * (30 + 25 + 1.25) = 734.5 msec

3. 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* = 30,000 EMPLOYEE records of *fixed length*. Each record has the following fields: Name (30 bytes), Ssn (9 bytes), Department_code (9 bytes), Address (40 bytes), Phone (10 bytes), Birth_date (8 bytes), Sex (1 byte), Job_code (4 bytes), and Salary (4 bytes, real number).

   a. Calculate the record size *R* in bytes.

      Record size = 30 + 9 + 9 + 40 + 10 + 8 + 1 + 4 + 4 = 115 bytes

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

      Bfr = floor(512/115) = 4

      Data blocks = roof(30,000)/4 = 7500

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. *Bfr;*

      index bfr = floor(512/(9 + 6)) = 34

   ii. the number of first-level index entries and the number of first-level index blocks;

      first-level index entries = 7500

      first-level index blocks = roof(7500/34) = 221

   iii. the number of levels needed if we make it into a multilevel index;

      3

   iv. the total number of blocks required by the multilevel index; and

      index blocks = 221 + 7 + 1 = 229

   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.

      3 + 1 = 4

d. Suppose that the file is *not ordered* by the key field Ssn and we want to construct a *secondary index* on Ssn. Repeat the previous exercise (part c) for the secondary index and compare with the primary index.

   i. the index blocking factor. *Bfr;*

      first level index bfr = floor(512/(9 + 7)) = 32
      above first level index bfr = floor(512/(9+6)) = 34

   ii. the number of first-level index entries and the number of first-level index blocks;

      first-level index entries = 30000

      first-level index blocks = roof(30000/32) = 938

   iii. the number of levels needed if we make it into a multilevel index;

      3

   iv. the total number of blocks required by the multilevel index; and

      index blocks = 938 + 28 + 1 = 967

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.

3 + 1 = 4

Note: unspanned organization means that each record is stored in only one block. In another words, don’t store part of a record in one block and the remaining part of the same record in another block.