Bilkent University

Computer Engineering Department

CS351 DATA ORGANIZATION AND MANAGEMENT

Midterm Date: November 26, 2011 Time: 13:40 – 15:30

Student Name/ ID No. Section	

GOOD LUCK!

Notes: 1. There are 100 points, 6 questions on 7 pages.

2. Please READ the questions. It is a closed book/notes exam.

3. Show your work and write the results do not leave them as an expressions.

4. You are not allowed to use your cell phone. You cannot share calculators

5. You cannot leave the exam room in the first 30 minutes.

6. Please do not write anything in the following table.

7. See below for the disk parameters to be used when appropriate.

Question No.	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Overall**
Points Possible	**20**	**12**	**16**	**22**	**10**	**20**	**100**
Your Grade							

DISK PARAMETERS: When/if needed use the following disk parameters:

B	block size	2400 bytes
btt	block transfer time	0.8 msec
ebt	effective block trans. time	1.0 msec
r	aver. rot. latency time	10 msec
s	average seek time	15 msec

1. *(20 pts.) Consider two unsorted sequential files F1 and F2. Each file contains 100,000 records and their record size is (R) is 400 bytes. In both files 20% of the records are marked as deleted and 40% of the active records of F1 and F2 are common. We want to find the difference between F1 and F2 in terms of active records, i.e., generate a new file*

F12= F1 - F2

the new file contains the active records that exist in F1 but not F2. For this purpose we have 5 MB byte of main memory allocated. The disk characteristics are defined on page 1.

The file difference program works as follows: we read 5MB of records of F1 to main memory and search the active records of F1 in F2, after processing all records of F1 stored in main memory we write the qualified records to the file F12 all together. This process is repeated until we are done.

Total file size = 400x100,000 = 40MB **Number of deleted records** = 0.2x100,000=20,000
Number of common active records = 0.4x(100,000-20,000) = 32,000
Number of uncommon active records = 0.6x(80,000) = 48,000

a*. How may (s+r) operation is needed for F1, F2, and F12?*

F1: 40MB/5MB = 8 We read F1 in 8 pieces, therefore number of (s+r) is **8**.

F2: For each active record of F1, we go to F2 and read it from its beginning. As F1 has 80,000 active records, number of (s+r) operation, needed for F2, is **80,000**.

F12: We write to F12 8 times, since we read F1 5MB at a time. Required number of (s+r) operation for F12 is **8**.

In the following two parts ignore (s+r) operations.

b*. How much time is needed to read the records of F1?*

We read the entire file only once (in 8 pieces). $b = no\ of\ blocks = \frac{100{,}000x400}{2400} = 16{,}667$
$b\ x\ ebt = 16{,}667\ x\ 1 = 16{,}667\ msec \cong \mathbf{16.7}\ sec$

c*. How much time is needed to read the records of F2?*

Each time, for each active record of F1, we read F2.
For common and active records, we read 0.5 (half) of the file.
For uncommon active records, we read the entire file.

No of common active records = 32,000 No of uncommon active records = 48,000
For common active records, we need $32{,}000 \times 0.5 \times 16.7 = 267{,}200\ \text{sec} = 74.2\ hours$
For uncommon active records, we need $48{,}000 \times 16.7 = 801{,}600\ \text{sec} = 222.7\ hours$
Total time is $74.2 + 222.7 \cong \mathbf{297}\ hours$

d*. How much time is needed to write records to F12?*

We write only the uncommon records, 48,000. Time needed to write the entire file is 16.7 sec.
Hence, $0.48\ x\ 16.7 \cong \mathbf{8}\ sec$

2. *(12 pts.) Consider a file of 90,000 records, record size is 100-bytes. (In this question merging is not considered.) The disk characteristics are defined on page 1. Explain each answer briefly.*

a. *How long does it take to sort this file using a single disk drive and 10 Mbytes of memory? Why?*

$90{,}000 \times 100 = 9\ MB$ => 9MB < 10MB

$b = \frac{90{,}000x100}{2400} = 3750$

$Time = 2 \times b \times ebt = 2 \times 3750 \times 1 = 7500\ msec = \mathbf{7.5}\ sec$

We use heap sort and have to read and write all blocks.

b. *How long does it take to sort this file using two disk drives and 10 Mbytes of memory? Why?*

In this question, we can assume that heap structure is 10MB.

Two disk drives imply replacement selection sort. In this case, $file\ size < memory\ size$.

There is no chance of overlapping read and write. Therefore, both sort and method requires the same amount of time.

$Time = 2 \times b \times ebt = \mathbf{7.5}\ sec$

If we choose heap structure less than 10MB, then we can use replacement selection sort with overlapping reads and writes.

$Time = (approximately) b \times ebt = \mathbf{3.75}\ sec$

Both solutions are accepted.

c. *How long does it take to sort this file using one disk drive and 2.5 MBytes of memory?*

The same as the (a) part. => **7.5** sec.

We have to read all blocks and write all blocks.

d. *How long does it take to sort this file using two disk drives and 2.5 MBytes of memory?*

We have two disk drives and $memory\ size < file\ size$ ($2.5MB < 9MB$)

We can overlap read and write.

Use replacement selection sort.

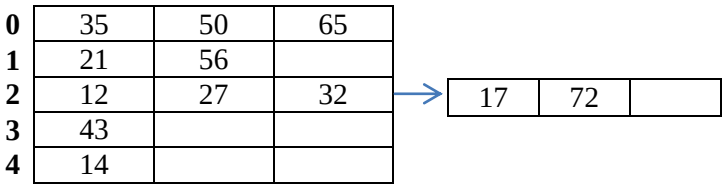
File processing time (approximately) $b \times ebt = 3.75$ sec

3. *(16 pts.) Consider a file with records of size 800 bytes (R= 800 bytes) The disk characteristics are defined on page 1. Using the hash function mod(key, 5) and block overflow chaining enter the following records (only the key values are shown) into an empty hash file.*

a*. Create the file for the following records.*

12, 27, 21, 35, 32, 14, 43, 56, 17, 50, 65, 77

Key	Mod(key,5)
12	2
27	2
21	1
35	0
32	2
14	4
43	3
56	1
17	2
50	0
65	0
77	2



b*. How much time is needed to access record 56? Why?*

1 disk access is needed. Note that both records (21 and 56) are in the main memory after read operation.

Therefore it is just one disk access.

$(s + r + btt) = 10 + 15 + 0.8 = \mathbf{25.8}\, msec$

c*. How much time is needed to access record 77? Why?*

2 disk accesses are needed; one for prime area and one for overflow area.

$2 \times (s + r + btt) = 2 \times 25.8 = \mathbf{51.6}\, msec$

4. *(22 pts.) Consider a linear hashing file environment. The current state of the file is as follows: It contains 50 primary area disk blocks; the boundary value is equal to 14; the current load factor of the file is 2/3. The desired load factor of the file is also 2/3. The blocking factor is 12. The following hash function is used to distribute the records among the blocks: Mod (key, 997).*

NOTE: We solve the question by correcting the bv. (Also see the next page.)

a. *What is the value of h?*
$n = 50 \implies \lfloor \log_2 n \rfloor = \lfloor \log_2 50 \rfloor = \mathbf{5}$

b. *What is the address (in binary) of the last page at level h?*
11111 ⇒ 31 in decimal
Note that $bv = (n-1) - (2^h - 1) = n - 2^h = 50 - 32 = \mathbf{18}$
(bv has to be corrected like this)

c. *How many disk blocks are at the hashing level h?*
$50 - 2 \times bv = 50 - 36 = \mathbf{14}$

d. *How many disk blocks are at the hashing level h+1?*
$2 \times bv = 2 \times 18 = \mathbf{36}$

e. *How many records are stored in the file?*
$$Lf = \frac{No.of\ records\ stored\ in\ file}{Prime\ area\ size} = \frac{2}{3} = \frac{no.of\ records}{50 \times 12}$$

$$\Rightarrow no.of\ records = \frac{50 \times 12 \times 2}{3} = 400$$

f. *Consider the query : Display record for record where Key= 1500. Which block do we need to access to answer this query?*

$mod(1500,997) = 503 \Rightarrow 111110111$

Use 5-suffix = $10111_2 = 23_{10} > bv \Rightarrow \boldsymbol{Look\ at\ page\ no.\ 10111_2\ or\ 23_{10}}$.

g. *Starting with the current state defined in the first paragraph how many records do we need to add to increase the value of bv (boundary value) by 1?*
$\frac{2}{3} \times 12 = \mathbf{8}$

h. *Again starting with the current state defined above how many records do we need to add to increase the hashing value by 1?*

Current load factor is $^2/_3$. We have 14 blocks at level h. Each time we add $^2/_3 \times 12 = 8$ records and increase the value of bv. We have 14 blocks at level h. $14 \times 8 = \mathbf{112}$.

i. *In this file environment what is the maximum number of prime area blocks that we can have in the file?*

Max number of prime area blocks can be **997**, because $0 \leq mod(key, 997) \leq 996$

4. *(22 pts.) Consider a linear hashing file environment. The current state of the file is as follows: It contains 50 primary area disk blocks; the boundary value is equal to 14; the current load factor of the file is 2/3. The desired load factor of the file is also 2/3. The blocking factor is 12. The following hash function is used to distribute the records among the blocks: Mod (key, 997).*

NOTE: We solve the question without correcting the bv.

a. *What is the value of h?* (The same as the previous solution.)
$n = 50 \implies \lfloor \log_2 n \rfloor = \lfloor \log_2 50 \rfloor = \mathbf{5}$

b. *What is the address (in binary) of the last page at level h?* (The same as the previous solution.)
11111 ⇒ 31 in decimal

c. *How many disk blocks are at the hashing level h?*
$50 - 2 \times bv = 50 - 28 = \mathbf{22}$ (or by using the formula 2^h - *bv*, there are 18 disk blocks)

d. *How many disk blocks are at the hashing level h+1?*
$2 \times bv = 2 \times 14 = \mathbf{28}$ (or by using the formula n - (2^h - *bv*), there are 32 disk blocks)

e. *How many records are stored in the file?* (The same as the previous solution.)

$$Lf = \frac{No.of\ records\ stored\ in\ file}{Prime\ area\ size} = \frac{2}{3} = \frac{no.of\ records}{50 \times 12}$$

$$\Rightarrow no.of\ records = \frac{50 \times 12 \times 2}{3} = 400$$

f. *Consider the query : Display record for record where Key= 1500. Which block do we need to access to answer this query?* (The same as the previous solution.)

$mod(1500,997) = 503 \Rightarrow 111110111$

Use 5-suffix = $10111_2 = 23_{10} > bv \Rightarrow \boldsymbol{Look\ at\ page\ no.\ 10111_2\ or\ 23_{10}}$.

g. *Starting with the current state defined in the first paragraph how many records do we need to add to increase the value of bv (boundary value) by 1?* (The same as the previous solution.)
$\frac{2}{3} \times 12 = \mathbf{8}$

h. *Again starting with the current state defined above how many records do we need to add to increase the hashing value by 1?*

Current value of bv is 14 and load factor is $^2/_3$. We have 22 blocks at level h. Each time we add $^2/_3 \times 12 = 8$ records and increase the value of bv. We have 22 blocks at level h.
$22 \times 8 = \mathbf{176}$ (or $18 \times 8 = 144$)

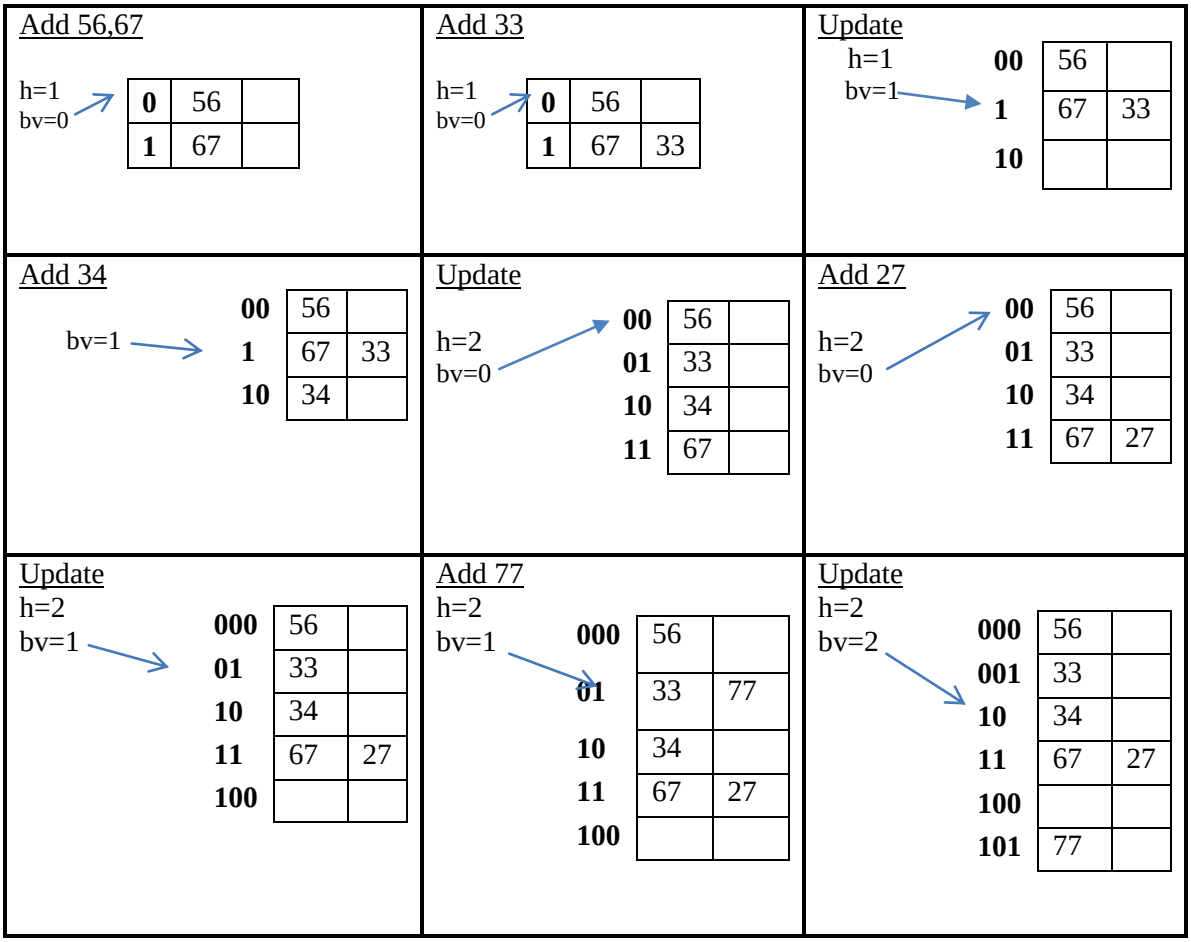
i. *In this file environment what is the maximum number of prime area blocks that we can have in the file?* (The same as the previous solution.)

Max number of prime area blocks can be **997**, because $0 \leq mod(key, 997) \leq 996$

5. *(10 pts.) Consider a linear hashing file environment with the desired Lf value= 1/2 and Blkf (Bucketing factor)= 2. Insert the following records (note that their pseudo key values are given) to this file beginning with an empty file. Show the changes in the file structure as you add the records.*

Pseudo Key (PK)	Binary Representation of PK
56	011 1000
67	100 0011
33	010 0001
34	010 0010
27	001 1011
77	100 1101

After reaching $Lf = {}^{1}/_{2}$, add 1 record and update the file.
Add 1 record and update.



6. *(20 pts.) Consider a file of size 700 MB. Assume that 5 MB of memory is available for sorting and merging. For sorting we want to use heap sort and for merging we want to use 4-way merge. The record size is 240 bytes. Disk characteristics are defined on the first page.*

a. *How many sorted segments are there after sorting?*

$$\frac{700}{5} = \mathbf{140}$$

b. *During merging how many passes are required? Draw a table that shows the merge pass number, size of each segment, and the number of segments?*

4-way merge ⇒ no. of passes $\lceil \log_4 140 \rceil = 4$

Pass No.	1	2	3	4
Size of each segment (MB)	5	20	8 x 80MB 1 x 60MB	2 x 320MB 1 x 60MB
No. of segments	140	35	9	3

c. *How much time is required in the first pass of merging excluding (s+r)?*

We need to read and write the entire file. $b = \frac{700 \times 10^6}{2400} = 291{,}667$

$Time = 2 \times b \times ebt = 2 \times 291{,}667 \times 1 = 583{,}333\ msec \cong \mathbf{583}\ sec$

d. *How much time is required in the final pass of merging excluding (s+r)?*

$Time = 2 \times b \times ebt \cong \mathbf{583}\ sec$

e. *How many physical and logical I/O is performed in the first pass of merging?*

Each block contains $\frac{2400}{240} = 10$ records.
We read all blocks. Each one is a physical read ⇒ $b = 291{,}667$
No of logical reads= $10 \times b \Rightarrow \mathbf{2{,}916{,}670}$

f. *How many physical and logical I/O is performed in the final pass of merging?*
The same as in part (e).