University of California, Berkeley
College of Engineering
Computer Science Division – EECS

Fall 2008 Prof. Michael J. Franklin

MIDTERM I

CS 186 Introduction to Database Systems

NAME:_____ **STUDENT ID:**_____

IMPORTANT: Circle the last two letters of your class account:

cs186 a b c d e f g h i j k l m n o p q r s t u v w x y z

a b c d e f g h i j k l m n o p q r s t u v w x y z

DISCUSSION SECTION DAY & TIME:_____ **TA NAME:** _____

This is a **closed book** examination – but you are allowed one 8.5" x 11" sheet of notes (double sided). You should answer as many questions as possible. Partial credit will be given where appropriate. There are 100 points in all. You should read **all** of the questions before starting the exam, as some of the questions are substantially more time-consuming than others.

Write all of your answers directly on this paper. **Be sure to clearly indicate your final answer** for each question. Also, be sure to state any assumptions that you are making in your answers.

GOOD LUCK!!!

Problem	Possible	Score
1. Buffer Manager	**25**	
2. B-Trees	**25**	
3. Hash Indexes	**24**	
4. Formal Query Languages	**26**	
TOTAL	**100**	

SID:_____

Question 1 – Buffer Management [3 parts, 25 points total]

a) [15 points] Consider a database system with three buffer frames (A, B, C) and a file of five disk pages (1, 2, 3, 4, 5). Assume that you start with an empty buffer pool. A sequence of requests is made to the buffer manager as described in the Request column (below). At certain times a Pin request is immediately followed by an Unpin request (represented as Pin/Unpin), but other times Pin and Unpin requests happen in an interlaced manner with other requests.

Fill in the following table showing the buffer contents after the completion of each operation using the **CLOCK** page replacement policy as described in the book and HW 1. For each page, indicate the pin count (PC) and for unpinned pages, indicate the value of the reference bit (true or false). You can mark unchanged buffer frames with "ditto" (").

Note: Total points for this problem is 15, (3 each for correct buffer contents at steps T4-T8)

Time	Request	Buffer Pages		
		A	B	C
T1	Pin 5	5 PC=1		
T2	Pin/Unpin 2	"	2 PC=0 Ref = t	
T3	Pin/Unpin 3	"	"	3 PC = 0 Ref =t
T4	Pin 4			
T5	Unpin 5			
T6	Pin/Unpin 3			
T7	Pin 1			
T8	Pin 2			

Question 1 – Buffer Management (continued)

b) [5 points] Consider an access pattern where there are multiple lookups of different keys in a B+tree. Assume that the B+Tree is 3 levels deep and the buffer pool is much larger than 3 frames. Which replacement policy will provide the best hit rate and why? **Circle your answer and be sure to state why.**

- **a)** LRU
- **b)** MRU
- **c)** Random
- **d)** All are equal

Why?

c) [5 points] Consider an access pattern where there are multiple lookups of different keys in a static hash file (with no directory). Assume that hash file is much bigger than the buffer pool. Which replacement policy will provide the best hit rate and why? **Circle your answer and be sure to state why.**

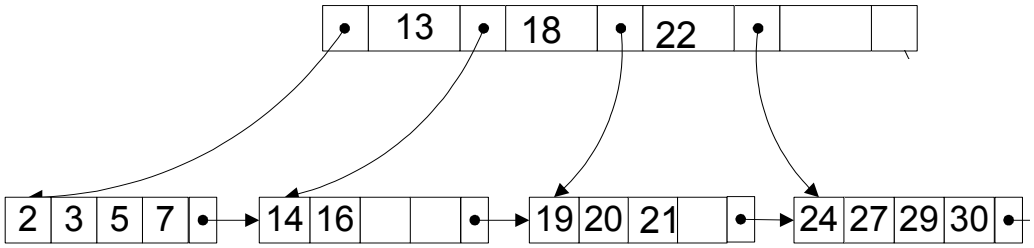
- **a)** LRU
- **b)** MRU
- **c)** Random
- **d)** All are equal

Why?

SID:_____

Question 2 – B+Trees [5 parts, 25 points total]

Consider the following B+Tree with order d=2



a) [7 points] Which of the following key values for data entries, if inserted on its own, would cause a leaf page split (Circle all that apply):

1 12 13 17 22 23 31

b) [2 points] Would any of the above inserts (done on its own) cause a non-leaf node to split? If so, which ones. If not, why not?

c) [2 points] Considering the original B+tree above, which of the following is true?

(**Circle one only, and briefly explain your answer)**

A. No deletions have ever been performed on it.

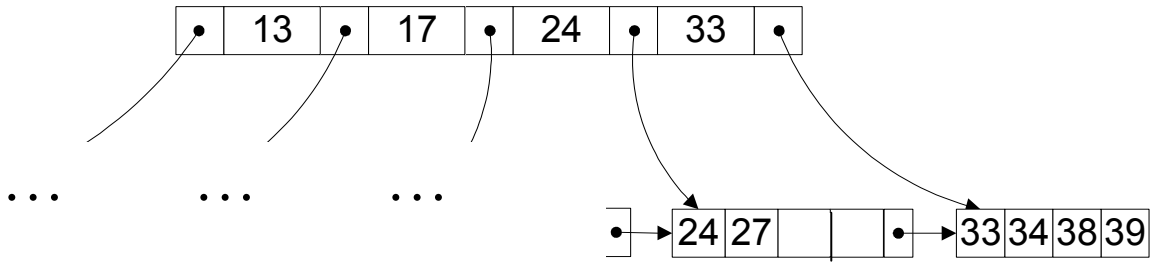
B. One or more deletions have been performed on it.

C. Not enough information to tell.

Why?

d) [4 points] STARTING WITH THE ORIGINAL TREE ABOVE, draw the B+Tree that results from deleting the data entry with the key 16. (you only have to draw the parts of the tree that change).

e) [10 points] Consider the following B+Tree with order d=2 (with some leaf nodes elided)



Draw the B+Tree that results from inserting a data entry with the key 40. (hint: recall that the root node is not required to have d keys in it).

Question 3 – Hashing [3 parts, 24 points]:

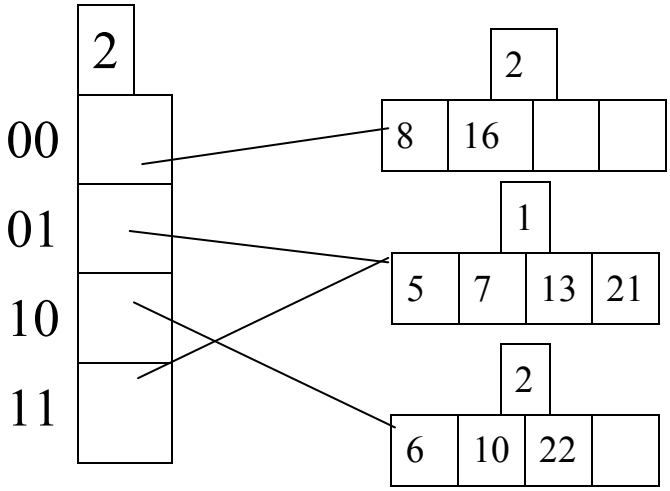
a) [12 points] Extendible Hashing

Consider the following 4 update operations.

operation no.	operation	key value (binary)
1	insert	25 (011001)
2	insert	46 (101110)
3	insert	18 (010010)
4	insert	20 (010100)

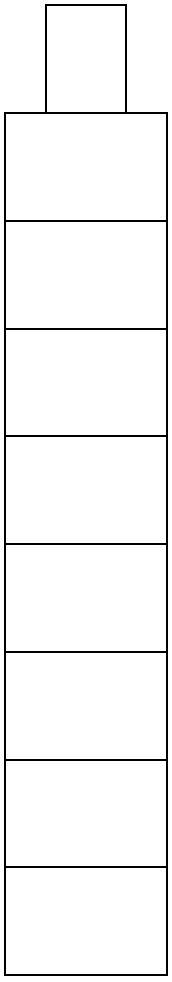
Now, consider an extendible hash structure where each bucket can hold up to 4 entries, with a hash function $h(n) = n \bmod 2^{depth}$ and an initial state as shown below.

Draw the extendible hash structure and its contents after the 4 operations have occurred in the order shown. We recommend that you do your scratch work on this page at first. But, this page will not be graded. You MUST put your final answer on the following page!!



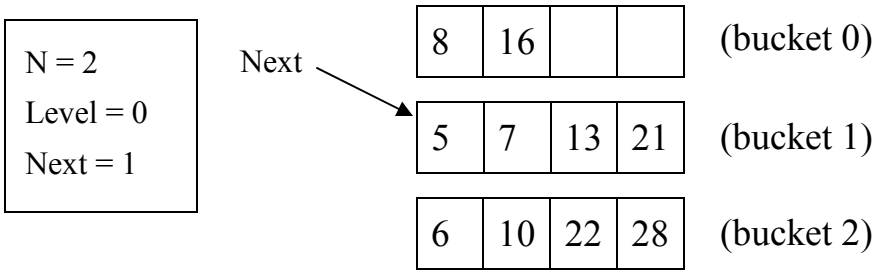
Final answer for Question 3(a) - Extendible Hashing:

- Only this page will be graded for question 3(a).
- The final structure should have a directory of size 8 so use the template below.
- Show all buckets and pointers
- Label the directory entries with their corresponding hash value (as on the previous page).
- Make sure to include local depths for all buckets and the global depth of the directory.



Question 3 – Hashing (continued)

Now, consider a *linear hashing* index with the following initial state:



Consider the sequence of inserts from part (a) shown in the table below and answer the following two questions (parts b and c).

operation no.	operation	key value (binary)
1	insert	20 (010100)
2	insert	46 (101110)
3	insert	18 (010010)
4	insert	23 (010111)

b) [6 points] Assuming the operations are performed in the order specified in the table, which of the four operations will cause the bucket pointed to by the "next" pointer (i.e., "bucket 1") to be split?

c) [6 points] What will be the value of "Next" after the split (from part b) happens (i.e., which bucket will be scheduled to be split on the subsequent overflow)?

Question 4 – Formal Query Languages [5 parts, 26 points total]

Consider the following schema that records information about a university (Primary keys are underlined):

Student (sid, sname, address, age)

Course (cid, cname, credits)

Professor(pid, pname, address, age)

Grade(sid, cid, pid, grade)

Write queries to answer the following questions. Note that we don't care about efficiency, but **points will be taken off for using unnecessary relations in your queries**. For each of the questions you can use your choice of Relational Algebra or Relational Calculus.

a) [3 points] Find the sid and sname of students who have received at least one A in a course.

b) [5 points] Find the sid and sname of students who have received at least one A in a *four-credit* course.

Recall the schema from the previous page (Primary keys are underlined):

Student (sid, sname, s_address, s_age)

Course (cid, cname, credits)

Professor(pid, pname, p_address, p_age)

Grade(sid, cid, pid, grade)

c) [6 points] Find the pids of professors who have taught **both** the course with cname "Databases" and the course with cname "Dance".

d) [6 points] Find the sids of students who have taken at least one course from a professor who is younger than they are.

e) [6 points] Find the sids of students who have taken a course from *every* professor.