EECS 484 (Fall 2016): Homework 4

Due Date: Tuesday Nov. 29th, by 11:59PM

Instructions:
For this homework, you may submit answers as many times as you like. Note however that only your last submission before the deadline will be counted.

If you are working in a group, both partners should submit individually, but declare the project partner below.

Also, note that you must submit all answers in a single submission for them to be counted. Giving half the answers in a first submission and half in a second, separate submission will not result in the best answer from each being taken. So be sure to save the link from your submission so that you can go back and edit your answers later, if necessary.

This homework counts towards the 10% HW grade (equal to other homeworks).

While it may appear you have extra time, note that there will be no office hours on Wednesday evening to Sunday (over the Thanksgiving break). And you have another homework and a project coming up. I recommend starting on the homework early.

Your email address (**zth@umich.edu**) will be recorded when you submit this form. Not **zth**? Sign out

If you are working in a group, please give the uniqname of your partner:
Leave blank if working alone. Note that you may work either alone or in groups of 2 at most. No sharing of work or solutions beyond your partner (Honor Code).

Part 1: (10 points) ARIES Protocol

Consider the following recovery log on the disk at the time of the crash. Assume that before step 1 in the LSN, the Transaction Table and Dirty Page Table are empty. Assume there none of the dirty pages are forced out to disk during this log.

LSN. LOG

1. T1 writes to P1
2. T2 writes to P2
3. T3 writes to P3
4. T1 writes to P1
5. T2 writes to P2
6. T1 commit
7. T1 end
8. begin checkpoint
9. end checkpoint (along with the dirty page table and transaction table)
10. T3 writes to P1

11. T3 commit
12. T3 end
13. T2 writes to P3
14. CRASH

(assume the system recovers fully after this crash)

Q1. (1 point) At which LSN will the ANALYSIS phase start processing the log?

- LSN 1
- LSN 3
- LSN 8
- LSN 10
- None of the above

Q2. (1 point) The LSN value from which the REDO phase will start processing the log is:

- 1
- 2
- between 3 and 7
- 10
- None of the above

Q3. (1 point) How many write operations to data pages will occur during REDO?

- 3
- 4
- 5
- 6
- None of the above

Q4. (1 point) How many CLRs would be added to the log during recovery?

- 0
- 1
- 2
- 3
- More than 3

Q5. (3 points) Consider a change in the problem specs: Assume that the amount of available memory to cache the pages is only two pages. When the system needs to evict a page, it evicts the least recently used page. For example, when P3 needs to be written at step 3, P1 will be evicted since P2 is more recently used. Otherwise, the problem specification is the same as earlier. With this change, figure out the Dirty Page Table (DPT) that would have been saved in the checkpoint and answer this question and the next question, based on the DPT. Which of the following pages will be marked dirty in the Dirty Page Table that will be saved with the checkpoint?

- P2 only
- P2 and P3 only
- P1 and P2 only
- P1 and P3 only
- None of the above

Q6. (3 points) (Follow-up) From what LSN value will the REDO phase start?

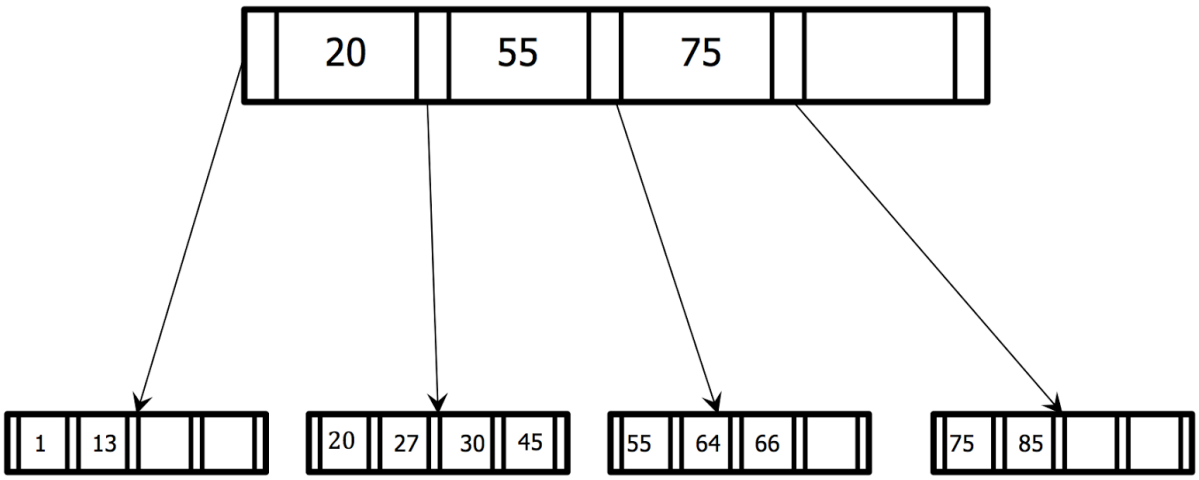
- 1
- 2
- 3
- 4
- None of the above

Part 2: B+ Tree Operations (20 points)

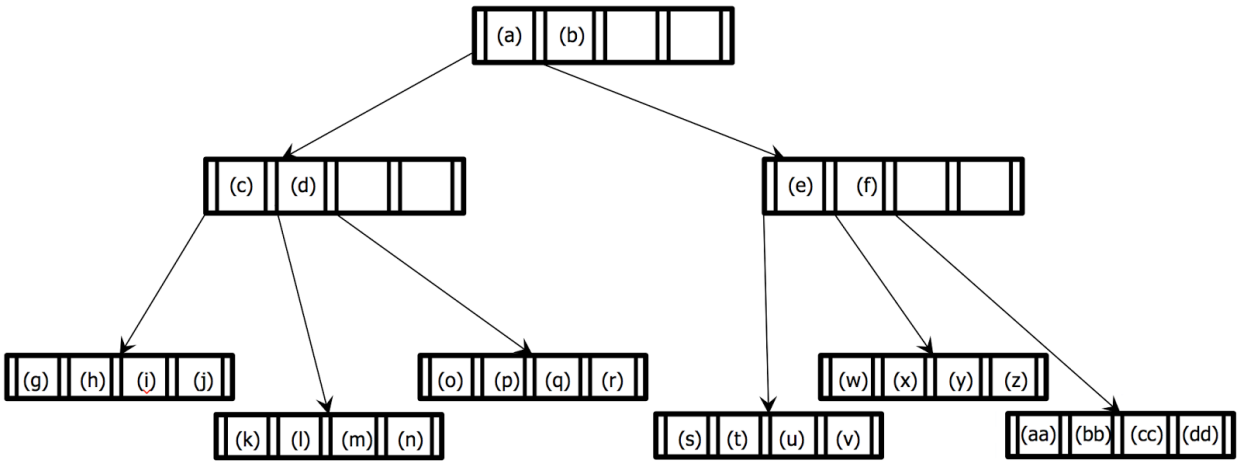
Consider the initial B+ tree below, and assume the convention that the left pointer points to values that are strictly less than the key value. Each node in this B+ tree can hold up to four entries (i.e., the order of the tree is 2).

For inserts, assume that when a node is split, the resulting right node will have equal or more values than the resulting left node. You should also assume that the value that is pushed up from the split for the index node is the smallest of the values from the right child. This is the same convention used in the lecture notes.

When borrowing (or redistributing) for question 2, make the two nodes as equal as possible, with { # entries in the left node } <= { # entries in the right node }. Also, when borrowing is possible from both the left sibling and the right sibling, you should favor the left sibling. If borrowing from the left sibling is not possible (or there is no left sibling), consider the right sibling. When borrowing, if the parent node's key requires a change, use the largest value possible for that key. If borrowing is not possible from either sibling, favor merging with the left sibling.



Question 1)



Using the template above, fill in the values based on how the B+ tree should look after inserting the values 6, 33, 69, 72 and 95 (in that order).

For this question, assume that the insert algorithm does NOT redistribute entries. Type 'none' if an entry should not have an associated value. You can ignore the single text box immediately below. Anything entered in it will have no impact on your grade.

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)

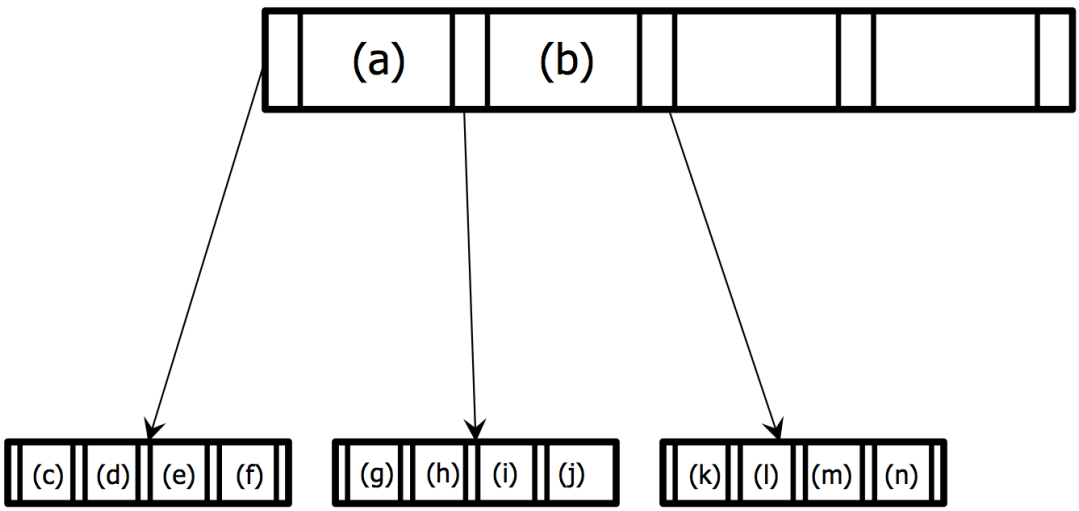
aa)

bb)

cc)

dd)

Question 2)



Using the template above, fill in the blanks to show how the original B+ tree for this problem will look after removing the values 1, 75, 20, and 64 (in that order).

For this question, assume that the delete algorithm considers redistributing entries. Type 'none' if an entry should not have an associated value. You can ignore the single text box immediately below. Anything entered in it will have no impact on your score.

a)

b)

c)

d)

e)

f)

g)

h)

i)

j)

k)

l)

m)

n)

Part 3: B+ Trees and Clustering (10 points)

Consider a table STUDENTS, where there is a B+ tree on attribute GPA, and the following query:

SELECT * FROM STUDENTS ORDER BY GPA;

Furthermore, assume that the first level (root) of the B+ tree is kept in memory at all times, and that the B+ tree being used has the following properties:

• height of the B+ tree: 3 (this is the length of any path from the root node to a leaf node in the tree)
• # of index leaf pages: 400
• # of data record pages: 6,000
• # of tuples (data records) per page: 50

For parts (a) and (b), provide the worst-case I/O cost of executing the above query by using the B+ tree (in terms of the number of page reads and writes) when:

a) The index is clustered

b) The index is unclustered

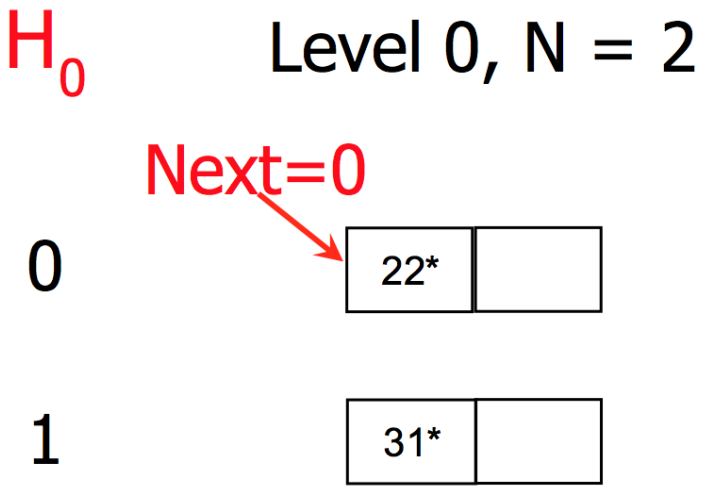
Part 4: Linear Hashing (14 points)

Consider the linear hashing index shown below. Assume that we use the same algorithm as discussed in lecture.

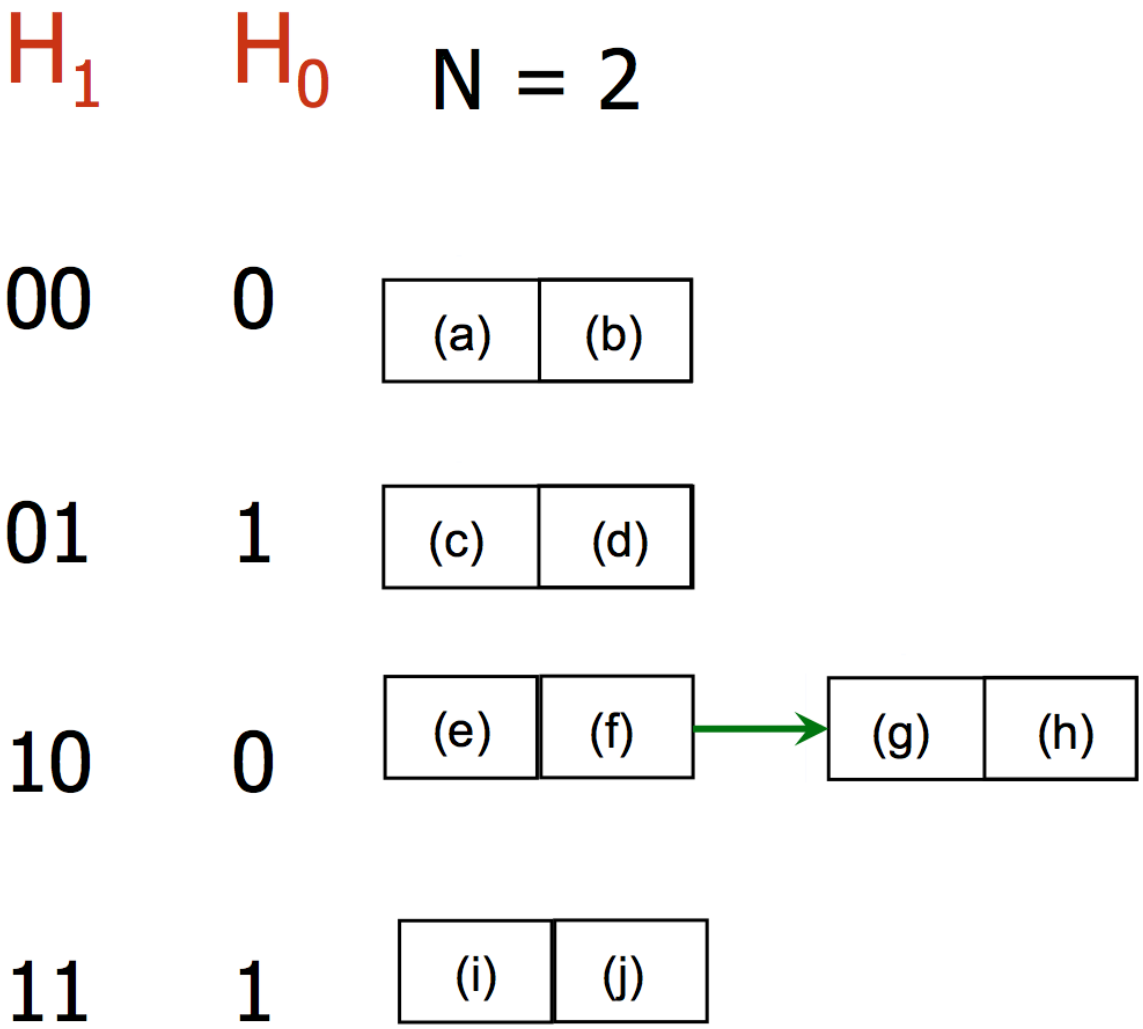
The hash function used is:

$$H_i(\text{value}) = \text{value mod}(2^i N),$$

where N is the initial number of buckets. Assume the bucket capacity is 2.



Question 1)



Suppose that we insert the values 43, 36, 21, 26, and 54 (In that order). Use the template provided above to show how the index will look afterwards.

Type 'none' if an entry should not have an associated value. You can ignore the single text box immediately below. Anything entered in it will have no impact on your score.

a)

b)

c)

d)

e)

f)

g)

h)

i)

j)

Which bucket is the Next pointer pointing to in this final state?

Use the H1 value to identify the correct bucket.

- 00
- 01
- 10
- 11

What is the Level in this final state?

Before You Submit

Know that once you click submit, you will receive a link that will allow you to return and edit your submission afterwards. You may continue to edit your submission up until the homework deadline, on Tuesday Nov.29th, at 11:59PM. Also, do not do a submission after the deadline. Any submission after that will result in a score of 0 as we may not have a copy of your submissions before the deadline. Only your last submission will be graded.

While you should also receive an email with the link to edit your submission, we recommend saving or bookmarking the link after submitting, just as a precaution.

Send me a copy of my responses.

Submit

Never submit passwords through Google Forms.

100%: You made it.