Carnegie Mellon University
Department of Computer Science

15-415/615 - Database Applications

C. Faloutsos, A. Pavlo , Spring 2015

Homework 4 (by Hong Bin Shim)

Due: **hard copy**, at **1:30pm, Feb. 26, 2015**

VERY IMPORTANT: Deposit **hard copy** of your answers, in class. Please

1. **Separate** your answers, on different page(s) for each question (staple additional pages, if needed).
2. **Type** the full info on **each** page: your **name**, **Andrew ID**, **course#**, **Homework#**, **Question#** on each question.

Reminders

- *Plagiarism*: Homework is to be done **individually**.
- *Typeset* all of your answers, please.
- *Late homeworks*: Standard policy: email (a) to all TAs, (b) with the subject line exactly `15-415 Homework Submission (HW 4)`, and (c) the count of slip-days you are using.

For your information:

- Graded out of **100** points. **6** questions total. Expected effort: ≈ 3-4h.
- **Solutions** for odd numbered exercises are available on the web: `http://pages.cs.wisc.edu/~dbbook/openAccess/thirdEdition/solutions/ans3ed-oddonly.pdf` You are strongly encouraged to use them.

Revision : 2015/02/19 04:46

Question	Points	Score
Buffers	16	
B Tree	20	
B+ Tree	20	
Extendible Hashing	16	
Linear Hashing	16	
Sorting	12	
Total:	100	

Question 1: Buffers .... [16 points]

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

Consider a buffer that can hold up to 4 frames and files numbered from 1 to 6 inclusive. Total of 8 requests come in to fetch files in this order: 1, 2, 3, 2, 1, 4, 5, 6. Consider three different buffer replacement policy: least recently used (LRU), most recently used (MRU), and clock.

(a) [**6 points**] For each policy, which page is evicted at which time as requests come in? Fill in the table below. (Put a "-" (dash) if nothing is evicted at certain timestamp)

timestamp	*request*	*evicted (LRU)*	*evicted (MRU)*	*evicted (Clock)*
t1	P1			
t2	P2			
t3	P3			
t4	P2			
t5	P1			
t6	P4			
t7	P5			
t8	P6			

(b) [**4 points**] Draw the final state of the buffer for each policy.

LRU:				
MRU:				
Clock:				

(c) [**6 points**] For each of the three policies, draw the final states of the buffer for a sequential scan of pages numbered from P1 to P10 inclusive. (Scanning 10 pages from page 1 to page 10, in order)

LRU:				
MRU:				
Clock:				

Question 2: B Tree .... [20 points]

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

Produce the most dense possible B-tree of order $d = 1$, containing as keys the integers 1 through 8 inclusive.

(a) i. [**3 points**] How many nodes does the structure have?

ii. [**4 points**] Draw the final structures.

(b) Produce the most sparse B-tree of order $d = 1$, containing the keys 1 through 7 inclusive.

i. [**3 points**] How many nodes does the structure have?

ii. [**4 points**] Draw the final structures.

(c) [**6 points**] Consider an empty B tree of order $d = 2$. Using the standard B-tree algorithm given in the foils (2-to-1 split, no deferred splits), insert keys from 1 to 10 (inclusive), in order. Draw the final structure.

Question 3: B+ Tree . . . . [20 points]

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

Consider the following B+ tree (This is Figure 10.29 of the textbook (Ed 3., p.367)). For each part, draw the tree after the specified operation.

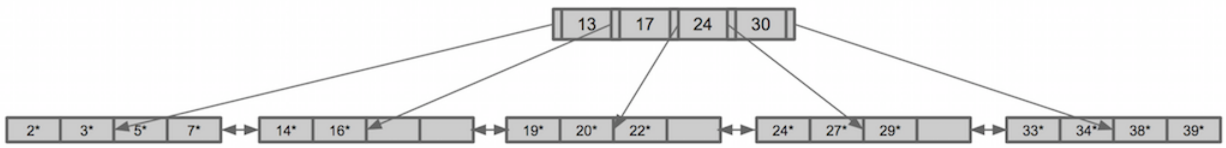

Figure 1: B+ tree.

- For each part of the problem, disregard previous parts and apply the instruction on the tree structure in Figure 1.
- With respect to "$\geq$", follow the convention used in the textbook, that is, the left pointer is for $<$, the right one for $\geq$.

(a) [**3 points**] Insert 17*.

(b) [**5 points**] Starting from the B+tree of Figure 1, insert 31*.
In case of merge, if you can borrow from both siblings, choose the one on the right.

(c) [**4 points**] Starting from the B+tree of Figure 1, delete 7*.

(d) [**4 points**] Starting from the B+tree of Figure 1, delete 24*.

(e) [**4 points**] Starting from the B+tree of Figure 1, delete 14*.

Question 4: Extendible Hashing .... [16 points]

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

Consider the following extendible hashing structure. (Figure 11.14 of the textbook (Ed. 3, p.387)), also repeated as Figure 2 for your convenience.

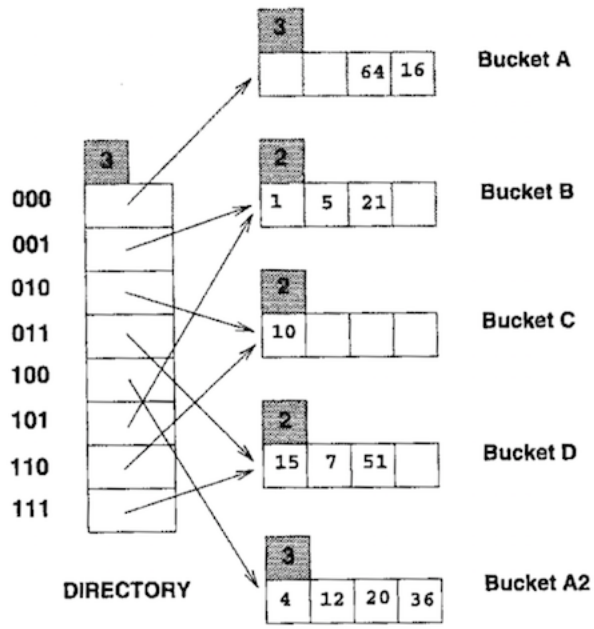

Figure 2: Extendible hashing.

- Unless explicitly specified, every sub-question is to be done on the hash table of Figure 2.
- Notice: small discrepancy between foils and textbook - please use the textbook convention, as does Figure 2. Specifically, the textbook has that keys in the same bucket, agree on the *suffix* of their hash value; in the foils, they agree on the prefix.

(a) [**4 points**] Which buckets will be split if you insert the following records in order: 26, 27, 28, 19?

(b) Starting from Figure 2, we want a record (positive integer) that is smaller than 36 and that will cause Bucket A2 to split.

i. [**4 points**] Give such an integer

ii. [**4 points**] Show the contents of the two resulting buckets after the split

(c) [**4 points**] Starting from Figure 2, how many buckets will remain after deleting the following records in order: 16, 64?

Question 5: Linear Hashing .... [16 points]

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

Consider the following linear hashing structure.

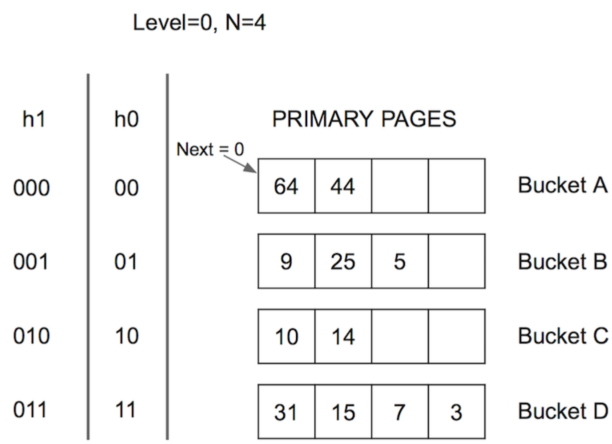

Figure 3: Linear Hashing

- This is a modified version of Figure 11.16 of the textbook (Ed. 3, p.389) with added entry '14' next to 10. Notice that the exercise is similar to Exercise 11.9, in the book.
- Feel free to check the solutions of the odd-numbered exercises, on the web.
- Assume that a bucket split occurs whenever an overflow page is created.

(a) [**4 points**] What is the *maximum* number of data entries that can be inserted in the best-case scenario, before you have to split a bucket?

(b) [**4 points**] Starting from Figure 3, draw the file after inserting the smallest possible single record whose insertion causes a bucket split.

(c) [**4 points**] Starting from Figure 3, which bucket would 128 be inserted into?

(d) [**4 points**] Starting from Figure 3, what is the smallest positive integer that can be inserted into the bucket A?

Question 6: Sorting .... [12 points]

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

You are sorting a file with N pages and you have B buffer pages. Consider the following scenarios and give the total I/O cost for sorting, in each scenario. (Use the algorithm shown in p.427 of the textbook.)

- HINT: This is a modified version of Exercise 13.1 in the textbook.
- Feel free to study the answers to the odd-numbered exercises, that are on the web.

(a) [**4 points**] A file with $N = 100,000$ pages and B=3 available buffer pages.

(b) [**4 points**] A file with N = 500,000 pages and B=7 available buffer pages.

(c) [**4 points**] A file with N = 3,000,000 pages and B=20 available buffer pages.