UMass Lowell
Department of Computer Science
Spring 2018

Instructor: Dr. Cindy Chen

COMP.5740 Midterm 1

March 6, 2018

2.5 hours
Closed book, closed notes

Name: Chuong Vu

Student ID: 01498233

Problem	**Score**	
1	(19%)	9
2	(19%)	15
3	(62%)	43+3
Total	(100%)	67+3=70

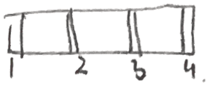
NOTE: Please write clearly.

Problem 1

2, 4, 6, 19, 41, 45, 54, 71, 73, 80

Construct a B+-tree for the following search-key values:

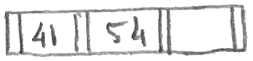
9



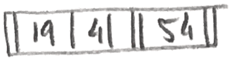
```
41, 54, 19, 71, 2, 45, 73, 80, 4, 6
```

Assume that the tree is initially empty and values are added in the order shown. Also assume that the number of pointers that will fit in one node is four. Show step by step results.

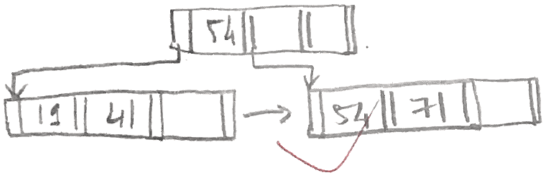
1.)



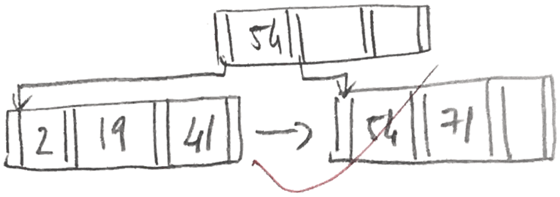
2.)



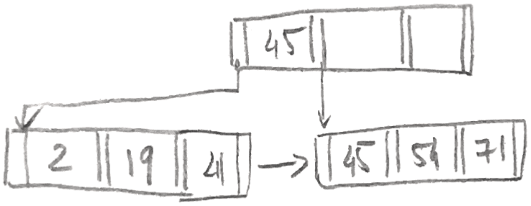
3.)



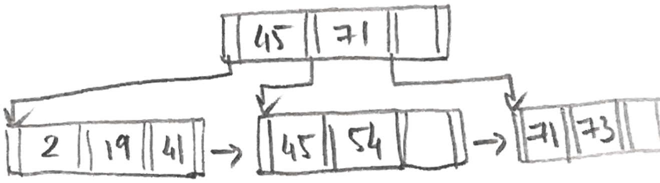
4.)



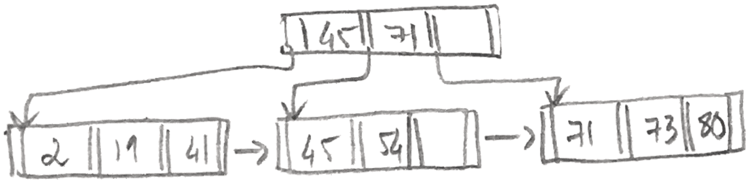
5)



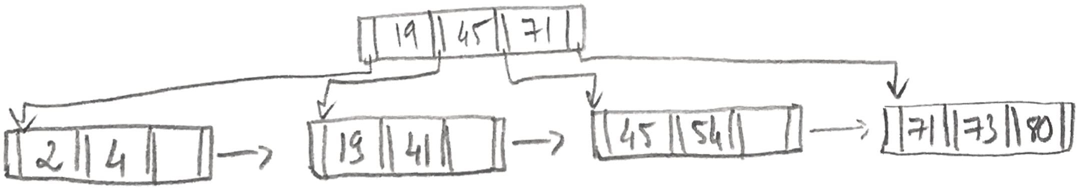
6)



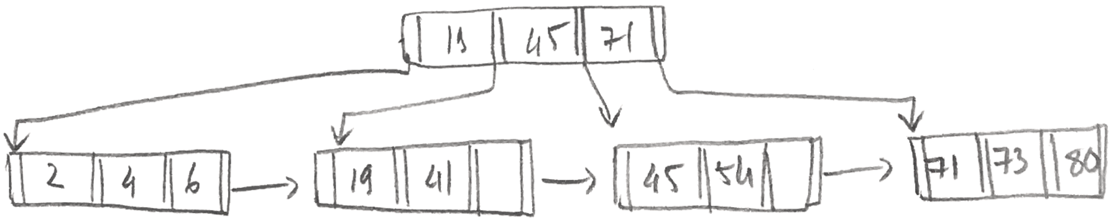
7)



8)



9)



Problem 2

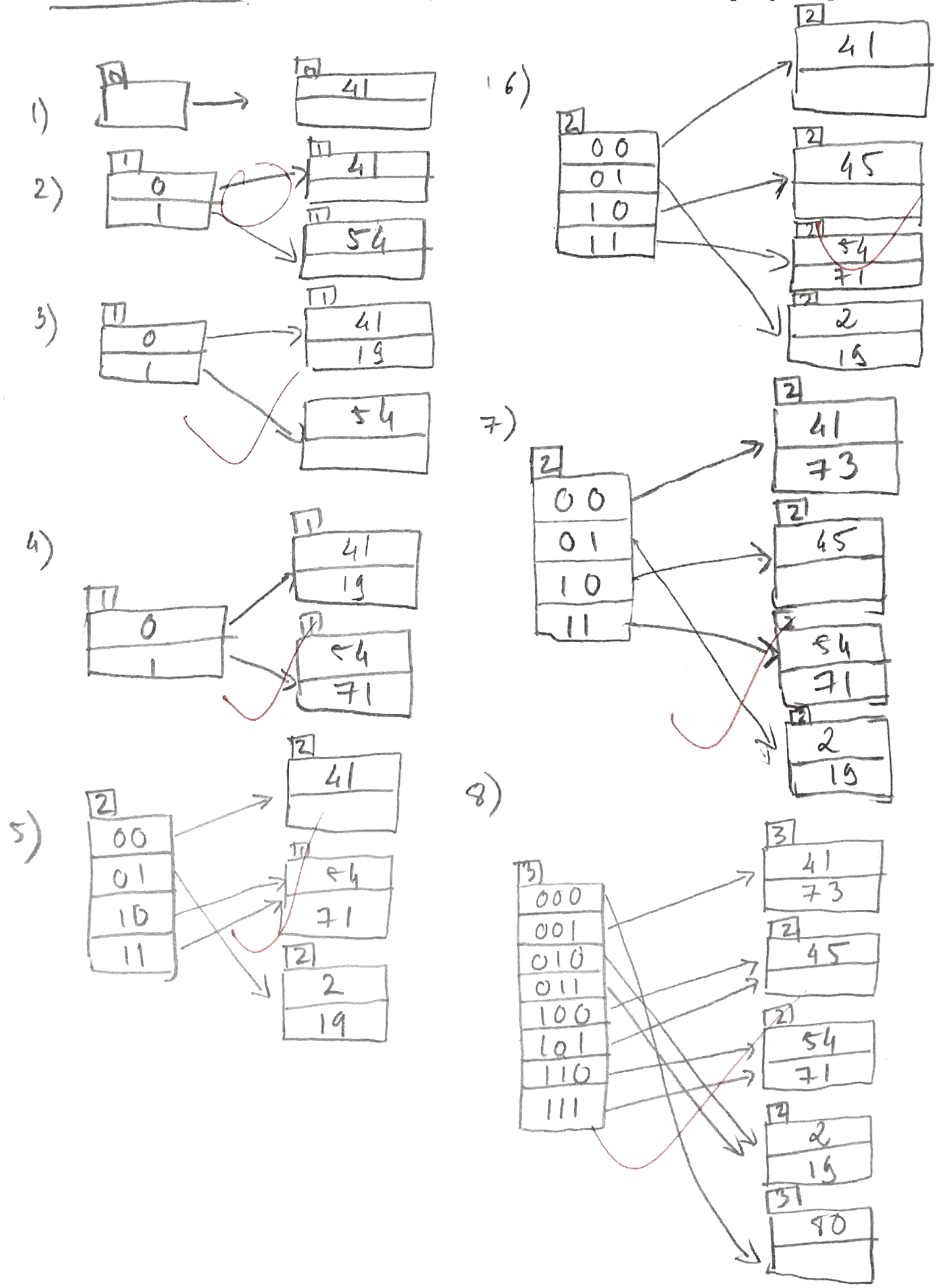
Suppose that we are using extendable hashing on a table that contains records with the following search-key values:

15

```
41, 54, 19, 71, 2, 45, 73, 80, 4, 6
```

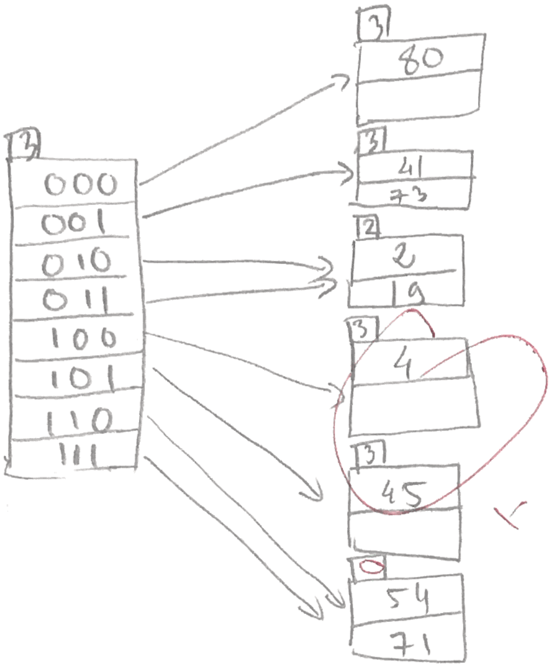
Construct the extendable hash structure using the hash function $h(x) = x \ mod \ 8$. Assume that the hash index is initially empty and values are added in the order shown. Also assume that each bucket can hold two records. Start from the left most bit. Show step by step results.

		X mode 8	
1) ✓	41	1	001
2) ✓	54	6	110
3) ✓	19	3	011
4) ✓	71	7	111
5) ✓	2	2	010
6)	45	5	101
7)	73	1	001
8)	80	0	000
9)	4	4	100
10)	6	6	110

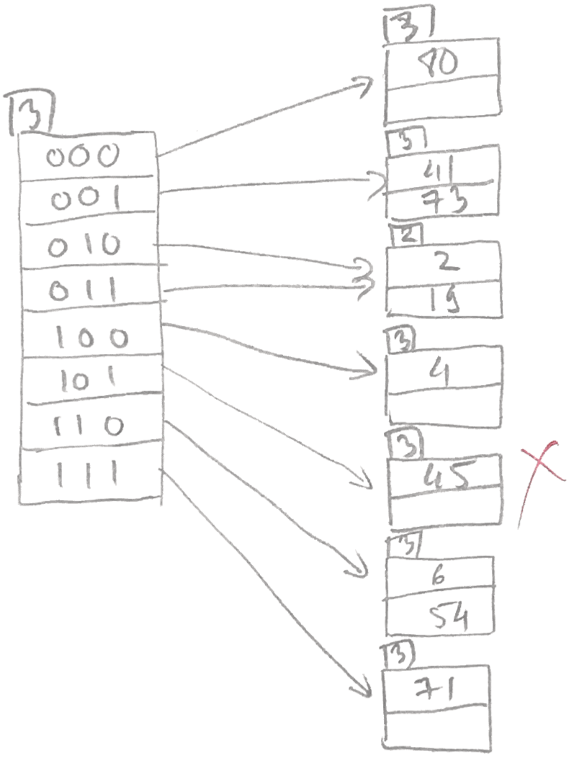


→ Next page.

9)


3
000
001
010
011
100
101
110
111
3
80
3
41
73
2
2
19
3
4
3
45
0
54
71

10)


3
000
001
010
011
100
101
110
111
3
80
3
41
73
2
2
19
3
4
3
45
3
6
54
3
71

Problem 3

Given the following relations describing:

```
student (sid, sname, dept, address, email, phone)
     course (cid, title, dept, credits)
  take (sid, cid, semester, year, score)
```

Assume the following:
(i) There are 70 pages for buffering in main memory. M = 70 pages
(ii) Each page holds 4096 bytes; each attribute is 20 bytes long. page size
(iii) The relation **student** has 100,000 tuples; the relation **course** has 50,000 tuples, the relation **take** has 4,000,000 tuples.
(iv) Two students may have the same name. sid will be unique → student is $\frac{100,000}{2} = 50,000$
(v) There are 10,000 departments. 10,000 unique department.
(vi) The range for score is 1 to 100.
(vii) The distribution of values is uniform.

Suppose that the following indexes exist on **student**: a primary B+-tree index on *dept*, and a secondary B+-tree index on *sname*.
Suppose that the following index exists on **course**: a hash index on *dept*.
Suppose that the following indexes exist on **take**: a primary B+-tree index on < *sid*, *score* >, and a secondary B+-tree index on *score*.

Answer the following questions. Show step by step results.

A (6 points) How many pages does each relation occupy on disk?

6

* Student:

$$\# \text{ of tuples/pages} = \left\lfloor \frac{4096}{6 * 20} \right\rfloor = 34$$

$$\# \text{ of pages} = \left\lceil \frac{100,000}{34} \right\rceil = 2942 \text{ pages.}$$

* Cource :

$$\# \text{ of tuples/page} = \left\lfloor \frac{4096}{20 * 4} \right\rfloor = 51$$

$$\# \text{ of pages} = \left\lceil \frac{50,000}{51} \right\rceil = 981 \text{ pages}$$

* take :

$$\# \text{ of tuples/pages} = \left\lfloor \frac{4096}{20 * 5} \right\rfloor = 40$$

$$\# \text{ of pages} = \left\lceil \frac{4,000,000}{40} \right\rceil = 100,000 \text{ pages.}$$

B (28 points) What is the minimal cost (in terms of numbers of pages transferred) of answering these queries?

sname from 1 to 2

(1) SELECT sid FROM student WHERE sname = "Joe"

2

Using B+ tree on sname

- # of pointer/node $= \left\lfloor \frac{4096 - 4}{20 + 4} \right\rfloor + 1 = 171$ ✓
- Height of B+ tree $= \left\lceil \log \frac{171}{2} * 20 \right\rceil = 1$
- # of qualifyng tuples $= \left\lceil \frac{100,000}{2} \right\rceil = 50,000$
- # of qualifyng pages = # of qualifyng tuples.

Cost · index secondary + data retrived.

$= 1 + 50,000 = 50,001$ pages

(2) SELECT cid FROM course WHERE dept = "CS"

7

Hash index on dept

- Cost is equal to hash table look up times number of qualifying tuples.
- # of qualifying tuples $= \left\lceil \frac{50,000}{10,000} \right\rceil = 5$
- Hash factor = 1.2

$\Rightarrow$ Cost = hash factor * # of qualifyng

$= 1.2 \times 5 = 6$ pages.

Secondary index

(3) SELECT sid FROM take WHERE score = 90 Using B+ tree on Score.

- # of pointer / node $= \left\lfloor \frac{4096 - 4}{20 + 4} \right\rfloor + 1 = 171$
- Height of B+ tree $= \left\lceil \log_{\frac{171}{2}} \times 100 \right\rceil = 2$
- # of qualifying tuples $= \left\lceil \frac{4,000,000}{100} \right\rceil = 40,000$
- # of qualifying pages = # of qualifying tuples.

  Cost = index secondary + data retrieved

  $= 2 + 40,000 = 40,002$ pages.

2nd index on score.

(4) SELECT sid FROM take WHERE score > 90

use <sid, sore>

+3

- # of pointer/node $= \left\lfloor \frac{4096 - 4}{20 + 4} \right\rfloor + 1 = 171$
- Height of B+ tree $= \left\lceil \log_{\frac{171}{2}} \times 100 \right\rceil = 2$
- # of qualifying leaf node $\left\lceil \frac{100 \times \frac{100 - 90}{100}}{\frac{171 - 1}{2}} \right\rceil = 1$
- # of qualifying tuples. $\left\lceil 4,000,000 \times \frac{100 - 90}{100} \right\rceil = 400,000$
- # of qualifying page $= 400,000$
- Cost of using B+ tree index

  $= (2 - 1) + 1 + 400,000 = 400,002$ pages.

C (28 points) What is the minimal cost (in terms of numbers of pages transferred) of joining the relations student and course using: Course ⋈ student

(1) Improved Block Nested Loop Join

$M = 70$

7

$$981 + \left(\left\lceil \frac{981}{70-2} \right\rceil * 2942\right) = 45,111$$

↑ course pages

↑ student pages.

student ⋈ course

$$2942 + \left(\left\lceil \frac{2942}{70-2} \right\rceil * 981\right) = 46,101$$

↑ student pages.

↑ course pages.

⇒ course ⋈ student will be better than student ⋈ course.

(2) Index Nested Loop Join course ⋈ student,

5

Use the primary B+ tree on dept in students.

- Height of B+ tree $= \left\lceil \log_{\frac{131}{2}} * 10,000 \right\rceil = 3$
- # of qualifying tuples $= \left\lceil 100,000 * \frac{1}{10,000} \right\rceil = 10$
- # of qualifying pages $= \left\lceil \frac{10}{70-2} \right\rceil = 1$
- Cost to join $= 981 + ((3 + 1) * 10,000) = 40,981$

$M = 70$

(3) Merge Join Course $\bowtie$ student.

7

+ Student is already sorted in sid.

+ Cost of sorting course table

$$981 * \left(2 * \left\lceil \log_{70-1} \frac{980}{70} \right\rceil + 1\right) = 2943$$

+ Cost of writing the sorted table to disk = 981

+ Total cost of merge join

$$2943 + 981 + (981 + 2942) = 7{,}847$$

(4) Hybrid Hash Join $m = 70$, $b_{course} = 981 \Rightarrow 70 \gg \sqrt{981}$ ✓

2

- Let n be # of partitions

$$\left\lceil \frac{b_s}{n} \right\rceil + 1 + n \leq m$$

$$n_1 = 20, \quad n_2 = 49.$$

Use $n = 2021$

- Partition course into 20 partition each has $\left\lceil \frac{981}{20} \right\rceil = 50$ pages.
- Partition student into 20 partition each has $\left\lceil \frac{2942}{20} \right\rceil = 148$ pages
- Cost $= 50 + 148 \cancel{*} (20 * [(981 - 50) + (2942 - 148)])$

$$(931 + 2794)$$

$$3725$$

$$= 74{,}69\cancel{8}$$