UCLA
Computer Science Department
Spring 2006

Instructor: J. Cho

Student Name and ID: [redacted]

CS143 Midterm: Closed Book, 90 minutes

(IMPORTANT PLEASE READ **):**

- There are 6 problems on the exam for a total of 60 points to be completed in 90 minutes. *You should look through the entire exam before getting started, in order to plan your strategy.*
- *Simplicity and clarity of your solutions will count.* You may get as few as 0 point for a problem if your solution is far more complicated than necessary, or if we cannot understand your solution.
- If you need to make any assumption to solve a question, please write down your assumptions.
- To get partial credits, you may want to write down how you arrived at your answer step by step.
- You may use one-page double-sided cheat-sheet during exam. You are also allowed to use a calculator.
- Please, write your answers neatly. Attach extra pages as needed. Write your name and ID on the extra pages.

Problem	Score	
1	10	10
2	10	10
3	10	10
4	10	9
5	10	10
6	10	7
Total	60	56

Exam Score:

Problem 1: 10 points

Consider the relation `Class(dept, cnum, title)`. The tuple ('CS', 143, 'Database') in the relation means that department 'CS' is offering class 143 on 'Database'.

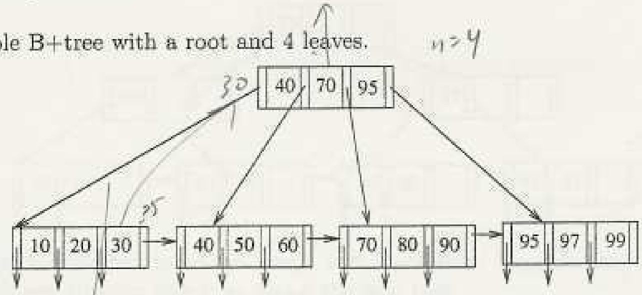
Write a relational algebra expression that returns all departments that offer at least three classes. Remember that simplicity and clarity count as well as correctness.

$$\pi_{dept}\left(\sigma_{\substack{c1.dept = c2.dept = c3.dept \\ \wedge \\ c1.cnum <> c2.cnum \\ \wedge \\ c2.cnum <> c3.cnum \\ \wedge \\ c1.cnum <> c3.cnum}}\left(\rho_{c1}(Class) \times \rho_{c2}(Class) \times \rho_{c3}(Class)\right)\right)$$

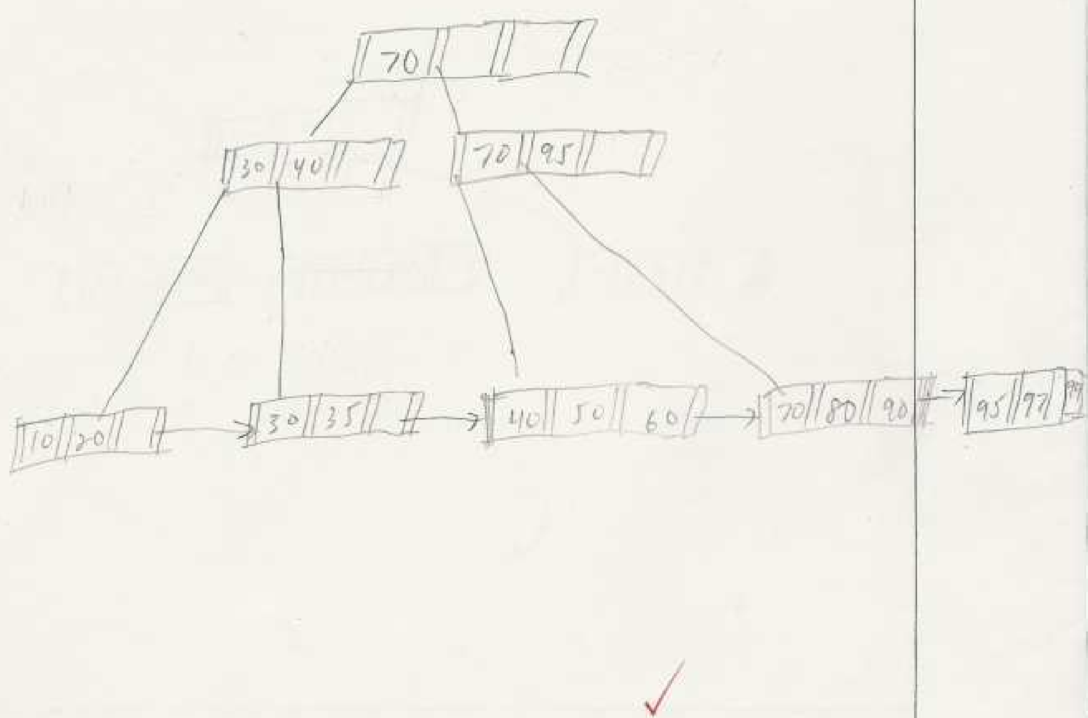
d1	c1	t1	d2	c2	t2	d3	c3	t3

Problem 2: 10 points

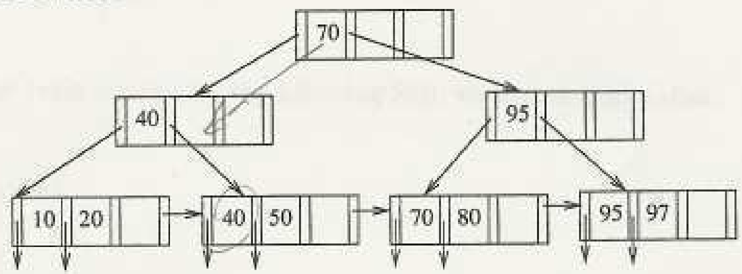
1. Here is a simple B+tree with a root and 4 leaves. n=4



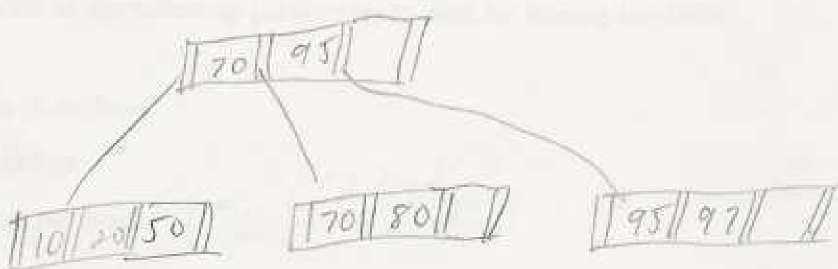
Assume that each node has room for 3 keys and 4 pointers. Also assume that when a node is split, and there are an odd number of pointers, the new node with the larger number of pointers is to the left of the other. Suppose that we insert a record with key 35 into this tree. Draw the new tree.



2. Here is another B+tree.



Assume that each node has room for 3 keys and 4 pointers. Suppose that we delete the record with key 40 from this tree. Draw the new tree.



Problem 3: 10 points

We want to store the table created by the following SQL statement into a disk.

```
CREATE TABLE Class(
  dept  CHAR(2),
  cnum  INTEGER,
  sec   INTEGER,
  unit  INTEGER,
  year  INTEGER,
  quarter INTEGER,
  title CHAR(30),
  instructor CHAR(20)
)
```

We need to store tuples for 1,000 classes that have been offered so far. 10 classes are offered every year. The tuples are stored in random order (i.e., they are not sequenced by any attribute). A disk of the following parameters is used for storing the table.

- 3 platters (6 surfaces)
- 1,000 cylinders
- 100 sectors per track $\left(\frac{10\,ms}{1\,track}\right)\left(\frac{1\,track}{100\,sectors}\right) = .1\,ms/sector$ = transfer time for each block
- 1024 bytes per sector
- 6,000 RPM rotational speed $\frac{6000\,rev}{60\,s} = 100\,rev/s \Rightarrow .01\,s/rev = 10\,ms/rev$
- 10ms average seek time

time to read 1 block = seek time + average rot latency + transfer time
$= 10\,ms + \frac{1}{2}(10\,ms) + .1\,ms = 15.1\,ms$

Assume that we have a B+tree on the year attribute and the tree has already been loaded into main memory. None of the disk blocks containing the Class table has been cached in main memory. Then what is the expected time to run the following query?

Since ∃ 10 classes in 2005, 10 tuples need to be retrieved.

```
SELECT * FROM Class WHERE year = 2005
```

Assume that each tuple is in different block and located in random location on disk.

$\boxed{151\,ms}$

✓

Time needed for IO is:

$10 \times (15.1\,ms) = 151\,ms$

Time to get addresses of blocks is negligible since B+ tree is in memory.

Problem 4: 10 points

R has n tuples
S has m tuples

Consider the relations $R(\underline{A}, \underline{B})$ and $S(\underline{B}, \underline{C})$. NULLs are not allowed for any attribute. Consider the relational algebra expression $\pi_B(R \bowtie \pi_B(S))$.

1. What is the minimum number of tuples that may result from the above expression? (3 points)

$\boxed{0}$

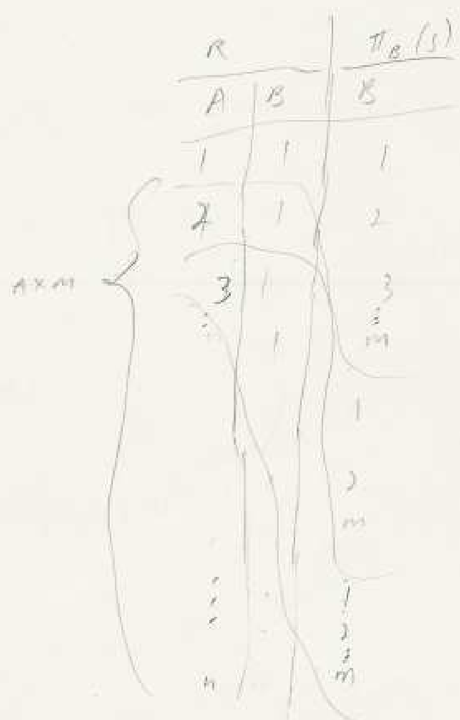
when R and S are independent tables (no key in one table references an attribute in the other)

2. What is the maximum number of tuples that may result from the above expression? (7 points)

$\boxed{\min(2n, m+n)}$

min(m,n)

-1



if $n > m$
$2m + (n-m) = m+n$

if $n < m$
$2n$

Problem 5: 10 points

A	B	
key	3	← delete
key	4	

Consider the following trigger on table $R(A, B)$.

```
CREATE TRIGGER ForgottenTrigger
AFTER INSERT ON R
  REFERENCING NEW ROW AS n
  FOR EACH ROW
WHEN ((SELECT COUNT(*) FROM R WHERE A = n.A) > 1)
BEGIN
  DELETE FROM R
  WHERE A = n.A AND B < SOME (SELECT B FROM R WHERE A = n.A);
END
```

remove duplicates (attribute "A" must be unique)

preserve tuple with largest B

Our database administrator should have created the above trigger before loading tuples to R, but he unfortunately forgot doing it and has already loaded tuples. Now your job is "fix" the table so that it contains only those tuples that would exist if the trigger had been created before loading the data.

Write a single SQL statement that does this job. Remember that simplicity and clarity count as well as correctness.

```
DELETE FROM R
WHERE A IN
    (SELECT A
     FROM R
     GROUP BY A
     HAVING COUNT (*) > 1)
AND B NOT IN
    (SELECT MAX (B)
     FROM R
     GROUP BY A
     HAVING COUNT (*) > 1)
```

✓

Problem 6: 10 points

Consider two SQL tables $R(\underline{A}, B)$, $S(A, \underline{B})$. NULL values are not permitted in any columns. $R.B$ is a foreign key referencing $S.B$.

```
Q1: SELECT A FROM R
    WHERE (B IN (SELECT B FROM S)) AND (A <> ALL (SELECT A FROM S))

Q2: (SELECT A FROM R) EXCEPT ALL (SELECT A FROM S)   bag semantics of Q3

Q3: (SELECT A FROM R) EXCEPT (SELECT A FROM S)   R.A − S.A

            R.A
Q4: SELECT A FROM R,S WHERE R.B = S.B AND R.A <> S.A
```

same based on given tables

Which of the following statements is true? $\boxed{3}$

1. We always get the same result from all four queries.
2. We may get up to two distinct results from the four queries.
3. We may get up to three distinct results from the four queries.
4. We may get up to four distinct results from the four queries.

If you choose any answer other than 1, show the *simplest* single instances of R and S that you can come up with that demonstrate the statement. In this case, show the result of all four queries on your instance of R and S in the space provided below.

−3

R

A	B
1	1
2	1

S

A	B
1	1
1	2

Q3: 2

Q4: 1

A	B
2	1
2	2