First Letter of your last name: Y

Your Full Name: Charles Yang

I promise I do not cheat on the exam. Your initial: CY

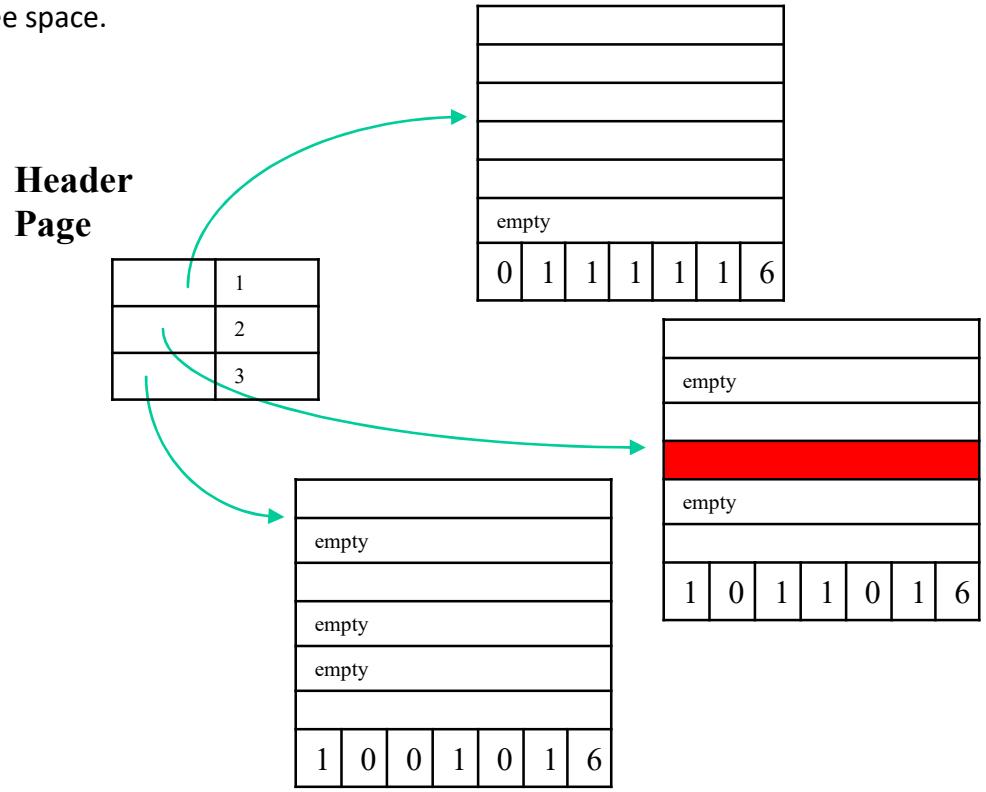
COM S 363: Exam 2

NOTES:

- This is an open book open note exam.
- You must work by yourself, without discussing with anyone else except TAs or instructors.
- You can handwrite, but you are strongly encouraged to type/draw on these slides directly. Some shapes may be given for your convenience. If you do handwrite, write legibly; if the grader can't read, at their sole discretion, you receive no point.
- Answer concisely and precisely. One effective way to lose points is to write nonsenses. HINT: The spaces you are given are more than you need.
- Submit your file in PDF format.

Problem	Max Points	Points
1	15	
2	25	
3	20	
4	15	
5	25	
Total	100	

1. (15 points) Suppose a relation is stored in heap files showing below. The slots with "empty" are free space.



(1) (4 points) From the illustration, can you decide if the records are fix length or variable length, packed or unpacked?

Unpacked, Fixed length

(2) (6 points) Suppose a user inserts another record. What the result files should look like?

The result files would look the same after insertion. The DBMS will search for empty space, and fill in one of the empty spaces. The corresponding bitmap value for where the record goes will become 1.

(3) (5 points) Suppose a user wants to delete a record. This record is located at the highlighted red slot. What is the procedure?

DBMS scans for the record of interest and removes it from the heap file.

2. (25 points) Consider a relation with this schema, Employee(ename:string, eid:integer, age:integer, salary:integer). Suppose ename is indexed by a sparse B^+-tree (primary) and eid is indexed by a dense and unclustered B^+-tree (secondary), as roughly illustrated by the figure showed in the next page. Suppose the relation has 1,000 pages, and each page is 512 bytes and can hold 16 records. Size of primary search key is 23 bytes, size of secondary search key is 2 bytes, and the size of a pointer is 2 byte.

   a) (5 points) Calculate the number of pages for data entry nodes in the secondary B^+-tree.

   1000 pages * 16 records/page = 16000 records
   16000 records * (2 keySize + 2 pointerSize) bytes/dataEntry = 64000 bytes
   64000 bytes / 512 bytes/page = 125 pages

   Estimate the total number of pages that need to be read from the disk in order to answer the following queries. Explain your answer. For each query, you can assume the selection factor is 0.1, i.e., out of 1,000 * 4 records, 1,000 * 4 * 0.1 = 400 records will satisfy each query condition.

   b) (5 points) Find all employees whose ages are in between 40 and 50.

   Search with Range Equality, Unsorted data
   Scan all 1,000 pages, I/O = 1000

   c) (7 points) Find all employees whose name starts with a character that is in between "C" and "F".

   Assume cost of traversing from root to leaf = 4 I/O

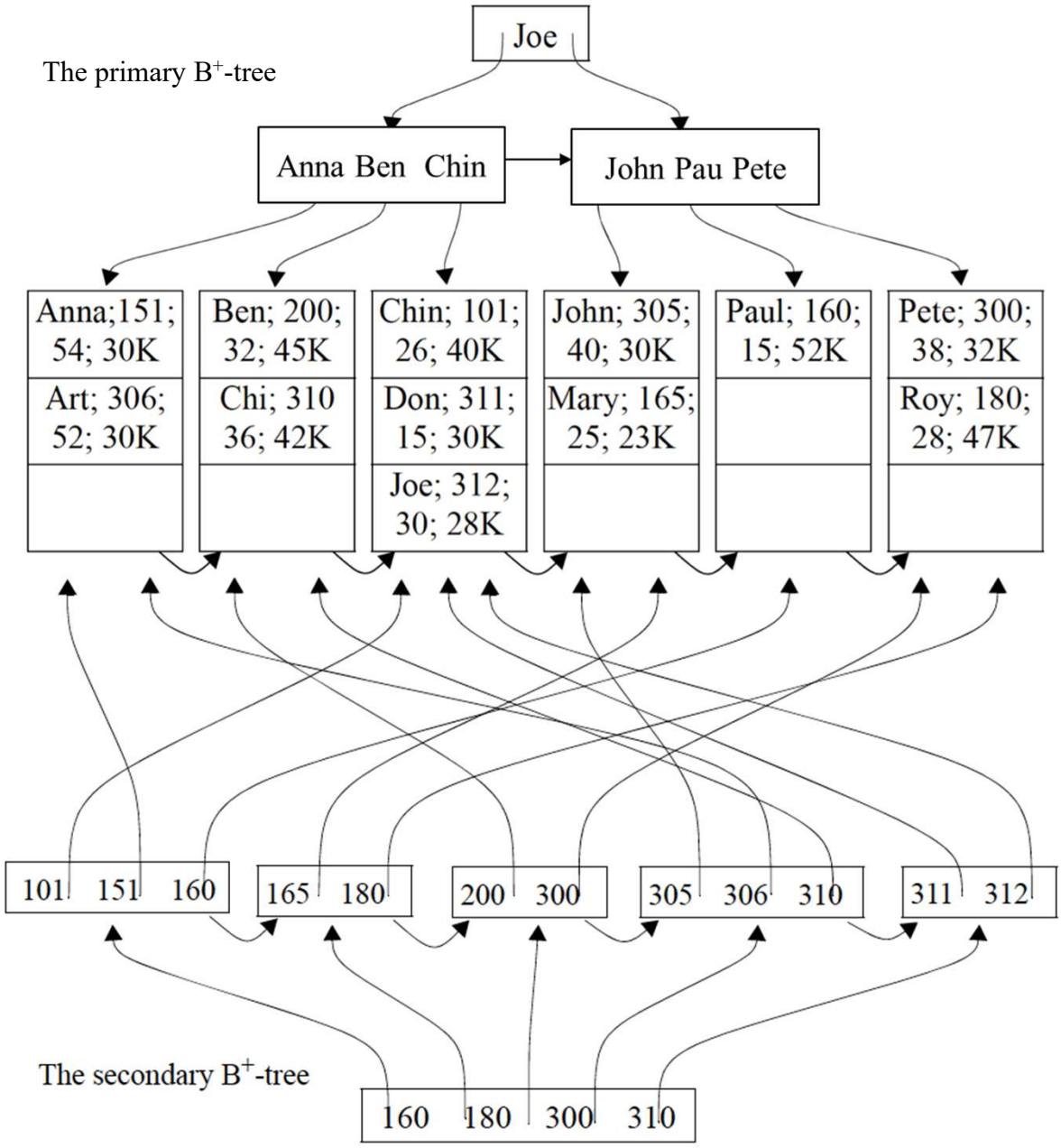
   Using primary key: keySize = 23, pointerSize = 2
   0.1 * 16000 records = 1600 matching records
   Data Entries Per Page = 512 / 25 = 20
   Pages w/ Matching Entries = 1600 / 20 = 80
   Pages w/ Matching Records = 1600 / 16 = 100

   4 + 80 + 100 = 184 pages I/O

   d) (8 points) Find all employees whose eid is in between 100 to 200.
   Using Secondary Key: keySize = 2, pointerSize = 2
   0.1 * 16000 records = 1600 matching records
   Data Entries Per Page = 512 / 4 = 128
   Pages w/ Matching Entries = 1600 / 128 = 13
   I/O for retrieving matching records = number of total pages = 1000 (unclustered)

   4 + 13 + 1000 = 1017

The primary B^+-tree



The secondary B^+-tree

3. (20 points) Explain how to sort 120 pages of data (e.g., numbers) with 4 pages of main memory in detail (i.e., pass 1, pass 2, ..., until the data is sorted) and estimate the I/O cost. Your algorithm needs to minimize disk I/Os (i.e., the number of pages that need to be read from and written to the disk). When sorting in main memory, you can assume you have the additional memory for swapping two numbers.

120 pages to sort, M = 4, assuming Read/Write = 1

Phase 1: Partition:

Read 120, Write 120, I/O = 120 + 120 = 240
120 / 4 = 30 subfiles

Phase 2: Merge Rounds

Merge every 3 subfiles into 1
Read 120, Write 120, I/O = 120 + 120 = 240
30 / 3 = 10 subfiles

Merge every 3 subfiles into 1
Read 120, Write 120, I/O = 120 + 120 = 240
10 / 3 = 4 subfiles

Merge every 3 subfiles into 1
Read 120, Write 120, I/O = 120 + 120 = 240
4 / 3 = 2 subfiles

Merge last 2 files
Read 120, Write 120, I/O = 120 + 120 = 240

240 * 5 = 1200 total estimated I/O for a multiway merge sort

4. (15 points) Suppose we have relation R of 120 pages in size, and relation S of size 40 pages, not sorted. We want to join R and S based on their primary keys. The size of main memory is 10 pages. Suppose the hash functions we choose uniformly partition the relation, fudge factor f=1.

   a) (6 points) After the partition phase, (numeric answers)

      R is divided into **9** partitions, each of which is about **14** pages;

      S is divided into **9** partitions, each of which is about **5** pages.

   b) (3 points) In the probing phase, which relation should be further partitioned using the B-2 memory pages?
      i. R
      ii. S
      iii. Both R and S
      iv. Either R or S
      v. Neither R nor S

      your answer: **i**

      (Note: iv is technically correct, as well)

   c) (3 points) Suppose the memory size is 5 pages instead of 10 pages. Is this change going to affect the I/O cost of Grace Hash Join? Choose your answer and explain why.
      i. Yes, I/O cost is going to increase
      ii. Yes, I/O cost is going to decrease
      iii. No, I/O cost is going to remain the same
      iv. It may or may not affect the I/O cost.

      your answer: **i**
      explanation: **R will require more partitioning.**

   d) (3 points) Suppose the memory size is 15 pages instead of 10 pages. Is this change going to affect the I/O cost of Grace Hash Join? Choose your answer and explain why.
      i. Yes, I/O cost is going to increase
      ii. Yes, I/O cost is going to decrease
      iii. No, I/O cost is going to remain the same
      iv. It may or may not affect the I/O cost.

      your answer: **ii**
      explanation: **R partitions can be accomplished in one step because each partition is 14 pages = 15 - 1**

5. (25 points) Derive the ***MINIMAL*** I/O costs of different join algorithms of relations R and S given the following variables, which you may or may not use all of them. Ignore the CPU time costs and the cost of writing the results. Write down steps for partial credits

   |R|=10: Number of tuples in R
   |S|=20: Number of tuples in S
   M=120: Number of pages in R
   N=40: Number of pages in S
   B=10: Number of available memory in pages

a) (5 points) What is the minimal I/O cost of block nested loop join?

S join R
40 + ceil(40/(10-2))*120 = 640

a) (5 points) What is the minimal I/O cost of simple nested loop join?

S join R
40 + 40 * 10 * 120 = 48040

a) (5 points) What is the minimal I/O cost of indexed nested Loops Join? (suppose the cost of retrieving a matching tuple is 2, for both R and S)

S join R
40 + 40 * 10 * 2 = 840

a) (5 points) What is the minimal I/O cost of grace hash join?

S join R. R join S
3(120 + 40) = 480

a) (5 points) What is the minimal I/O cost of Sort-Merge Join? (suppose the join is on their primary keys which are sorted already)

S join R. R join S
120 + 40 = 160