York University **Test One / Fall 2007**
Department of Computer Science and Engineering **CSE4411**

- This is a closed book, **75 minutes** test.
- **No questions are allowed during the test. If in doubt, write down your doubts and assumptions and proceed with your answer.**

LAST NAME	**SOLUTIONS**
FIRST NAME	**SOLUTIONS**
YORK ID#	**SOLUTIONS**
CS LOGIN	**SOLUTIONS**

Exercise 1 [10 points]
Does the final structure of a B+ tree depend on the order in which the items are added to it? Explain your answer and give an example.

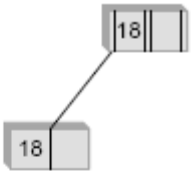
Answer:
Yes, because when an insertion causes a split of a leaf page, the entry to be inserted in the parent node depends on the order in which items have been added to the tree.

Exercise 2 [15 points]
Starting with an empty B+ tree with up to two keys per node (either for the index set or the sequence set) show how the tree grows when the following keys are inserted one after another:

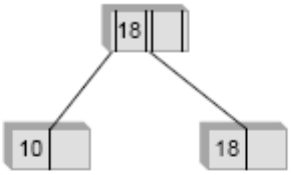
18, 10, 7, 14, 8, 9, 21

Answer:

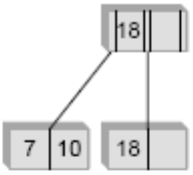
1. insert 18:



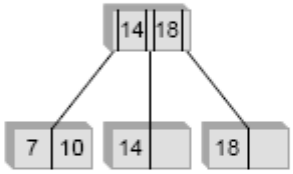
2. insert 10:



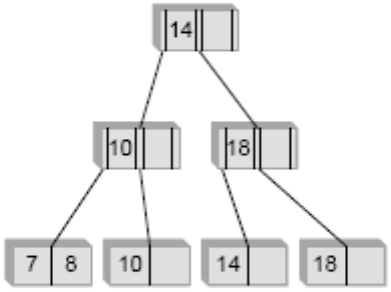
3. insert 7:



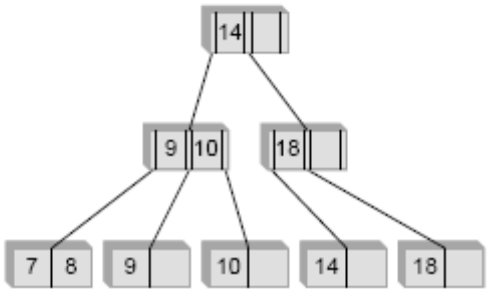
4. insert 14:



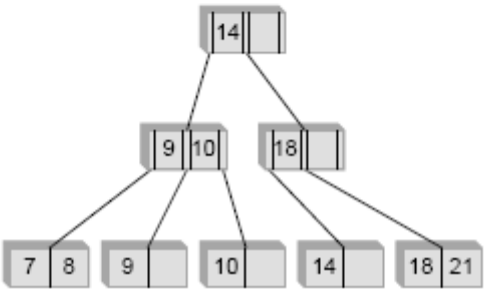
5. insert 8:



6. insert 9:



6. insert 21:



Exercise 3 [10 points]
Consider the following data and parity-block arrangement on four disks (assume some unconventional RAID architecture distribution):

Disk 1	Disk 2	Disk 3	Disk 4
B1	B2	B3	B4
P1	B5	B6	B7
B8	P2	B9	B10
…	…	…	…
…	…	…	…

Each Bi represents a data block. Each Pi represents a parity block. Parity block Pi is the parity block for data blocks B_{4i-3} to B_4. what, if any, problem might this arrangement present?

Answer:
This arrangement has the problem that P_i and B_{4i-3} are on the same disk. So if that disk fails, reconstruction of B_{4i-3} is not possible, since data and parity are both lost.

Exercise 4 [20 points]

A particular table in a relational database contains 100,000 rows, each of which requires 200 bytes of memory. Estimate the time in milliseconds to to insert a new row into the table when each of the following indices on the related attribute is used. Assume a page size of 4K bytes and a page access time of 20 ms.

a. [5 points] No index (heap file)
b. [15 points] A clustered, non-integrated B+ tree index, with no node splitting required. Assume that each index entry occupies 100 bytes. Assume that the index is 75% occupied and the actual data pages are 100% occupied. Assume that all matching entries are in a single page.

Answer:

a) Append (at the end of file). Just one IO, i.e., 20 ms

b) If we assume that each entry in the index occupies 100 bytes then an index page can hold 40 entries. Since the data file occupies 5000 pages, the leaf level of the tree must contain at least 5000/40, or 125 pages. Then the number of levels in the tree (assuming page 75% occupancy in the index) is $\lceil \log_{30} 125 \rceil + 1$ = 3. Assume that the index is clustered, not integrated with the data file and that all matching entries are in a single page, 4 I/O operations and 80ms are required to retrieve all matching records. Two additional I/O operations are required to update the leaf page of the index and the data page. Hence, the time to do the insertion is 120ms.

Exercise 5 [20 points]

(a) [10 points] Start with a directory size 2 (i.e. d = 1) and use extendible hashing to insert the following keys (in the order listed):

2, 3, 5, 7, 11, 17, 19, 23, 29.

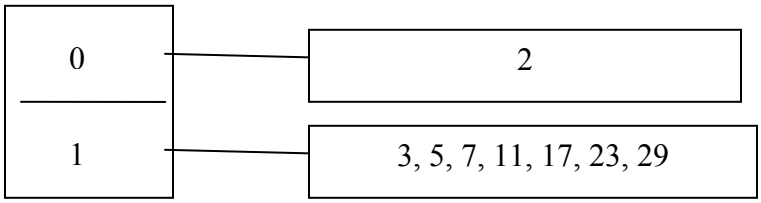
Use as your hash function h(k) = k mod size_of_directory – i.e., the d right most binary digits of the hash value. Show the table after each insertion (or if you feel confident that the resulting table is ok, show only the final table!)

(b) [5 points] After the insertions, delete 3 and 11. Show the table after the deletions.

(c) [5 points] Is it possible to combine any buckets after the deletions? If yes, explain why and show the resulting table (after combining of the buckets). If no, explain why.

Answer:

(a) – notice, there was no bucket capacity given, so there is no occurring bucket overflow.



(b) after deleting 3 and 11 the table is same as above except that bucket_1 does not have 3, 11.

(c) yes.