CSE4411 fall 2007

Assignment #1: Due October 24, 2007, 2 pm

Weight 2.5%

SOLUTIONS

This assignment involves working with file structures (chapters 8, 9, 10, 11 of textbook).

Exercise 1: do exercise 9.14 from the textbook.

Solution:

Because most page references in a DBMS environment are with a known reference pattern, the buffer manager can anticipate the next several page requests and fetch the corresponding pages into memory before the pages are requested. This is *prefetching*.

Benefits include the following:

1. The pages are available in the buffer pool when they are requested.
2. Reading in a contiguous block of pages is much faster than reading the same pages at different times in response to distinct requests.

Exercise 2: do exercise 9.20 from the textbook.

Solution:

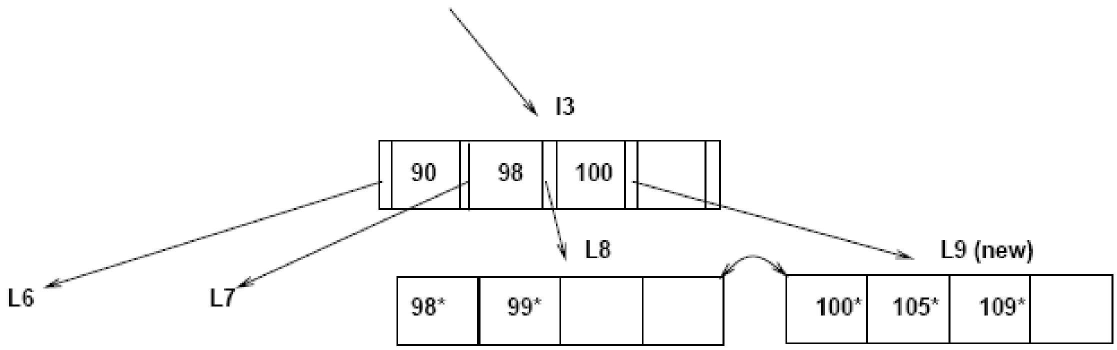
1. In the first approach (a list of all pages and a list of pages with free space) a page with free space belongs to both lists. Thus, we need to have one set of pointers (*previous* and *next*) per list, per page. In the second approach, each page belongs to exactly one of these lists, and it suffices to have a single pair of *previous* and *next* pointers per page. Other than this, the two approaches are quite similar. The second approach therefore, is superior overall.

2. This is outlined in the answer to the previous part.

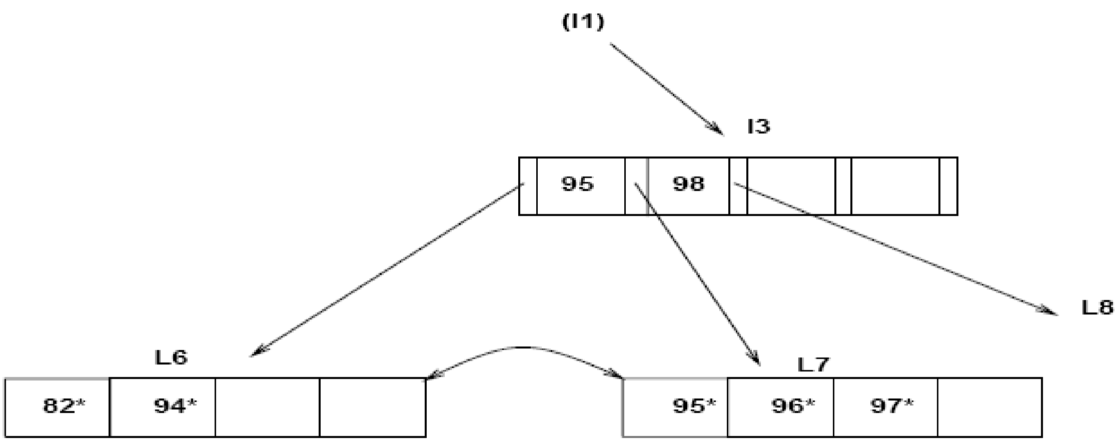
Exercise 3: do exercise 10.2, questions 1, 2, 3 from the textbook. Assume that rotations, when applicable, are done with respect to the right sibling.

Solution:

1. I1, I2, and everything in the range [L2..L8].
2. See Figure below. Notice that node L8 is split into two nodes.



3. Assuming that there is redistribution from the right sibling, the solution can be seen in the Figure below.



Exercise 4: do exercise 10.6, questions 1, 2 from the textbook.

Solution:

1. The answer to each part is given below

(a) Since ISAM trees use overflow buckets, any series of five inserts and deletes will result in the same tree.

(b) If the leaves are not sorted, there is no sequence of inserts and deletes that will change the overall structure of an ISAM index. This is because inserts will create overflow buckets, and these overflow buckets will be removed when the elements are deleted, giving the original tree.

2. The height of the tree does never change since an ISAM index is static. If a leaf

page becomes full, an overflow page is allocated; if a leaf page becomes empty, it remains empty.

Exercise 5: do exercise 11.7 from the textbook.

Solution:

Let *h* be the height of the B+ tree (usually 2 or 3) and *M* be the number of data entries per page (*M* > 10). Let us assume that after accessing the data entry it takes one more disk access to get the actual record. Let *c* be the occupancy factor in hash indexing.

Consider the table shown below (disk accesses):

Problem	Heap File	B+ Tree	Hash Index
1. All tuples	10^5	$h + \frac{10^6}{M} + 10^6$	$\frac{10^6}{cM} + 10^6$
2. $a < 50$	10^5	$h + \frac{50}{M} + 50$	100
3. $a = 50$	10^5	$h + 1$	2
4. $a > 50$ and $a < 100$	10^5	$h + \frac{50}{M} + 49$	98

1. From the first row of the table, we see that heap file organization is the best (has the fewest disk accesses).
2. From the second row of the table, with typical values for *h* and *M*, the B+ Tree has the fewest disk accesses.
3. From the third row of the table, hash indexing is the best.
4. From the fourth row or the table, again we see that B+ Tree is the best.