
Database Systems

Instructors: Hao-Hua Chu
Winston Hsu
Fall Semester, 2007

Assignment 4

Deadline: At the end of the class, Nov 28 (Wed), 2007

This is an individual assignment, that is, no group submissions are allowed.

Cheating Policy: If you are caught cheating, your grade is 0.

Late Policy: We will not accept any assignment submissions.

Questions

1. Consider the page format for variable-length records that uses a slot directory.
 - a. One approach to managing the slot directory is to use a maximum size (i.e., a maximum number of slots) and allocate the directory array when the page is created. Discuss the pros and cons of this approach with respect to the approach discussed in the text.
 - b. Suggest a modification to this page format that would allow us to sort records (according to the value in some field) without moving records and without changing the record ids.

2. Modern disk drives store more sectors on the outer tracks than the inner tracks. Since the rotation speed is constant, the sequential data transfer rate is also higher on the outer tracks. The seek time and rotational delay are unchanged. Given this information, explain good strategies for placing files with the following kinds of access patterns:
 - a. Frequent, random accesses to a small file (e.g., catalog relations).
 - b. Sequential scans of a large file (e.g., selection from a relation with no index).
 - c. Random accesses to a large file via an index (e.g., selection from a relation via the index).
 - d. Sequential scans of a small file.

3. Consider the B+ tree index of order $d = 2$ shown in the figure below.
- Show the tree that would result from inserting a data entry with key 9 into this tree.
 - Show the B+ tree that would result from inserting a data entry with key 3 into the original tree. How many page reads and page writes does the insertion require?
 - Show the B+ tree that would result from deleting the data entry with key 8 from the original tree, assuming that the left sibling is checked for possible redistribution.
 - Show the B+ tree that would result from deleting the data entry with key 8 from the original tree, assuming that the right sibling is checked for possible redistribution.
 - Show the B+ tree that would result from starting with the original tree, inserting a data entry with key 46 and then deleting the data entry with key 52.
 - Show the B+ tree that would result from deleting the data entry with key 91 from the original tree.
 - Show the B+ tree that would result from starting with the original tree, inserting a data entry with key 59, and then deleting the data entry with key 91.
 - Show the B+ tree that would result from successively deleting the data entries with keys 32, 39, 41, 45, and 73 from the original tree.

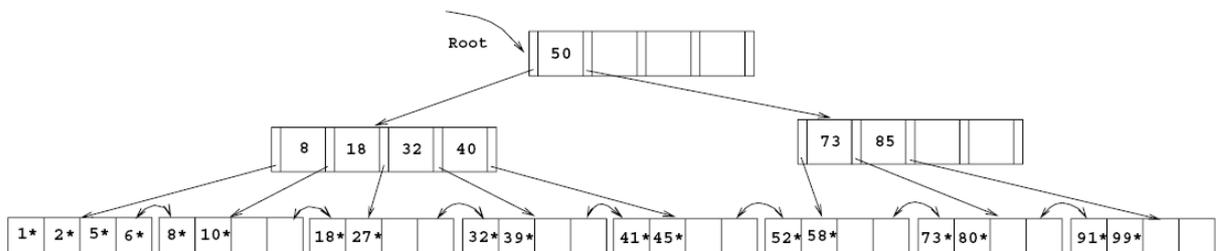


Figure for Question 3

4. Consider the B+ tree index shown in the figure below, which uses Alternative (1) for data entries. Each intermediate node can hold up to five pointers and four key values. Each leaf can hold up to four records, and leaf nodes are doubly linked as usual, although these links are not shown in the figure. Answer the following questions.
- Name all the tree nodes that must be fetched to answer the following query: "Get all records with search key greater than 38."
 - Insert a record with search key 109 into the tree.
 - Delete the record with search key 81 from the (original) tree.
 - Name a search key value such that inserting it into the (original) tree would cause an increase in the height of the tree.
 - Note that subtrees A, B, and C are not fully specified. Nonetheless, what can you infer about the contents and the shape of these trees?
 - How would your answers to the preceding questions change if this were an ISAM index?
 - Suppose that this is an ISAM index. What is the minimum number of insertions needed to create a chain of three overflow pages?

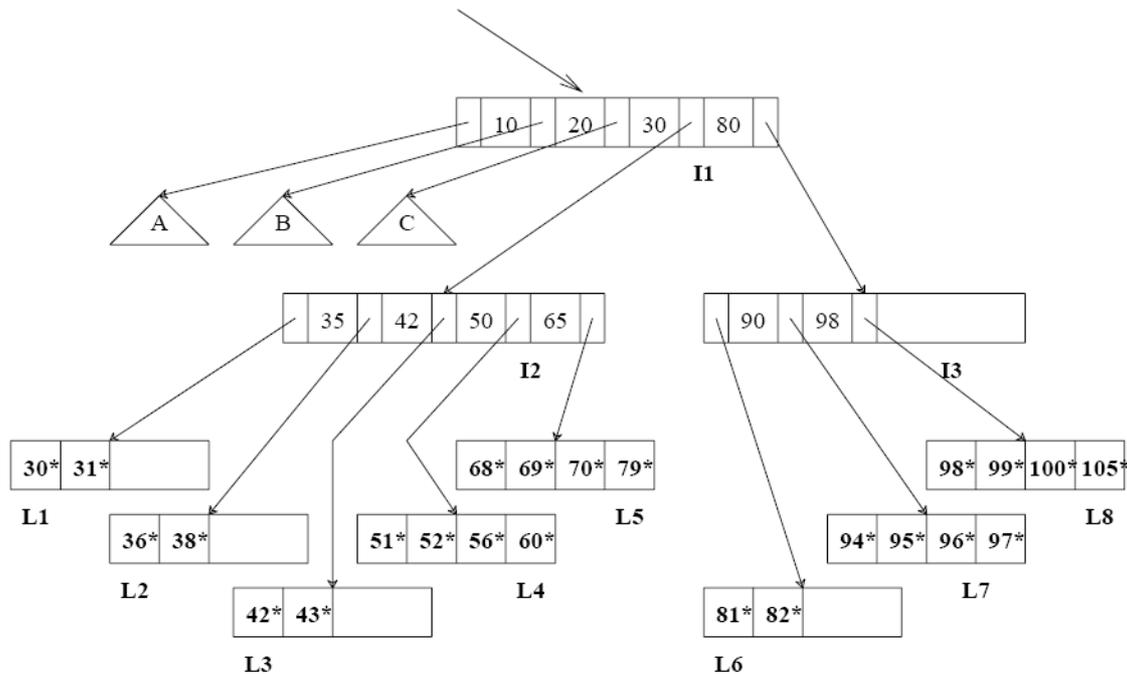


Figure for Question 4

Submission

Hand in PAPER PRINTOUT that contains your answers to the four questions. Please include your name and ID.