CPS216 Advanced Database Systems (Data-Intensive Computing Systems, Fall 2009), Assignment 2

- Due date: Tuesday, Oct. 13, 2009, in class (2.50 PM).
- Submission: In class, or email solution in pdf or plain text to shivnath@cs.duke.edu.
- Do not forget to indicate your name on your submission.
- State all assumptions. For questions where descriptive solutions are required, you will be graded both on the correctness and clarity of your reasoning.
- Email questions to shivnath@cs.duke.edu.

Question 1 **Points 5**

Consider the portion of the B-Tree shown in Figure 1. What is the permissible range of values for X and Y?

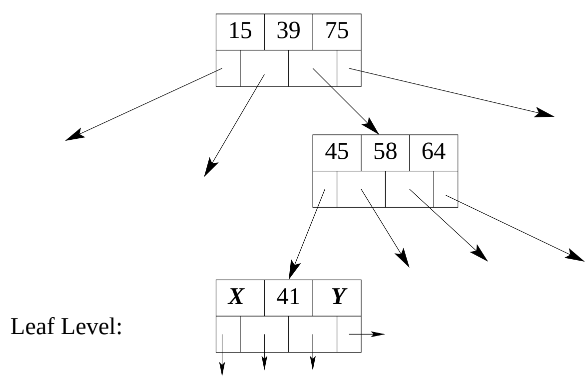

Figure 1: Part of a B-Tree with $n = 3$

Question 2 **Points 10**

Consider the B-Tree shown in Figure 2. What is the maximum number of keys that can be inserted into the B-Tree without necessitating the addition of a new level? Give an example key insertion sequence to support your answer.

Question 3 **Points 10**

What is the minimum number of key insertions that causes a new level to be introduced in the B-Tree of Figure 2? Give an example insertion sequence having the minimum number of keys that causes a new level to be added.

Question 4 **Points 5**

Consider the portion of the B-Tree shown in Figure 3. Delete key 62 and update the B-Tree so that only the three nodes shown in Figure 3 are modified. Show the state of the three nodes after the deletion.

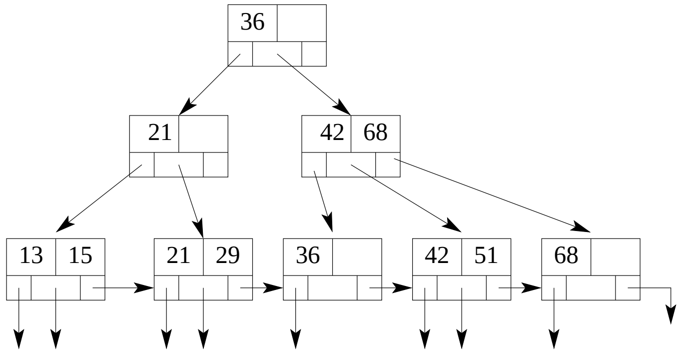

Figure 2: A B-Tree with $n = 2$

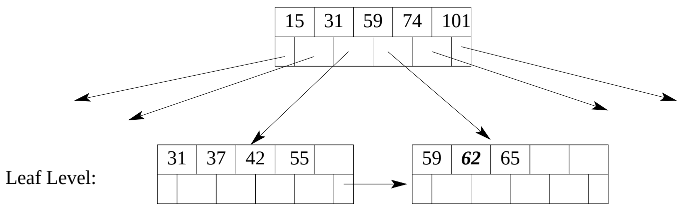

Figure 3: Part of a B-Tree with $n = 5$

Question 5 **Points 15**

Consider a B-Tree with parameter n. For each node there is a limit on the minimum number of pointers that the node can have. For internal nodes, the limit given in class is $\lceil \frac{n+1}{2} \rceil$, for leaf nodes the limit is $\lfloor \frac{n+1}{2} \rfloor$ (pointers to data), and for the root node the limit is 2 (assuming the B-Tree has at least 2 indexed keys). Indicate whether each of the statements (a)-(c) are true or false. Provide brief explanation.

(a) The limits for internal and leaf nodes can be reduced below $\lceil \frac{n+1}{2} \rceil$ and $\lfloor \frac{n+1}{2} \rfloor$, respectively.

(b) The limits for internal and leaf nodes can be increased beyond $\lceil \frac{n+1}{2} \rceil$ and $\lfloor \frac{n+1}{2} \rfloor$, respectively.

(c) For root node, the limit can be increased beyond 2.

Question 6 **Points 15**

Consider insertion and deletion operations over a B-Tree. Clearly, an insertion or a deletion operation changes the *state* of a B-Tree. By "state" we mean the exact set of nodes comprising the B-Tree, and the keys and pointers stored in these nodes. Assume there are no duplicate keys. Indicate whether each of the statements (a)-(c) are true or false. Provide brief explanation.

(a) Inserting a key k and immediately deleting it can leave the B-Tree in a different state.

(b) Inserting key k_1 followed by key k_2 always leaves the B-Tree in the same state as inserting k_2 followed by k_1.

(c) Consider the insertion of key k_1 followed immediately by the deletion of key k_2 ($k_1 \neq k_2$). The height of the B-Tree can increase during the insertion of k_1 and decrease during the deletion of k_2.

Question 7 **Points 15**

This question is about construction of B-Tree indexes. We covered two techniques for constructing B-Trees in the class: A *sort-based* technique, and an *insert-based* technique (which inserts the keys of the index one after another). Indicate whether each of the statements (a)-(d) are true or false. Provide brief explanation.

(a) The sort-based construction involves a very small number of random I/Os, and the bulk of the I/Os is sequential.

(b) The insert-based construction performs better than the sort-based one if the keys are inserted in sorted order.

(c) The sort-based construction always achieves a space utilization close to 100%.

(d) The insert-based construction always achieves a space utilization close to 100%.

Question 8 **Points 10**

The B-Tree shown in Figure 4 has duplicate keys. For example, there are two entries for key 43. Do you see any complications with the presence of duplicate keys in a B-Tree. What solution do you suggest to address these complications?

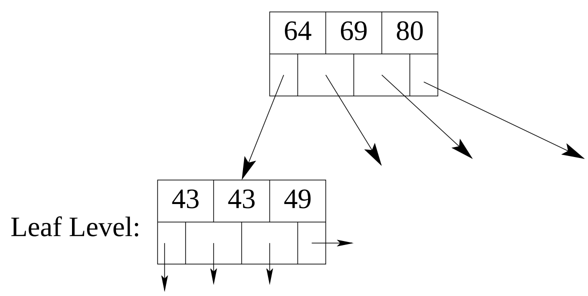

Figure 4: Part of a B-Tree with $n = 3$

Question 9 **Points 5**

Consider a 3.5 inch disk with 2 magnetic surfaces with 64 tracks per surface, rotating at 3600 rpm. It has a usable capacity of 2 megabytes (2×2^{20} bytes). Assume 20% of each track is used as overhead (gaps). Also, assume that the usable capacity is equally distributed among the tracks.

a. What is the burst bandwidth this disk can support?

b. What is the sustained bandwidth this disk can support?

c. What is the average rotational latency?

d. Assuming the average seek time is 16 ms, what is the average time to fetch a 2-kilobyte (2×2^{10} bytes) sector?

Question 10 **Points 10**

Consider a disk with the following properties:

- There are four platters providing eight surfaces.
- There are $2^{13} = 8192$ tracks per surface.
- There are (on average) $2^8 = 256$ sectors per track.
- There are $2^9 = 512$ bytes per sector.
- The disk rotates at 3840 rpm.

- The block size is $2^{12} = 4096$ bytes.
- Assume 10% of each track is used as overhead.
- The time it takes the head to move n tracks is 1 + n/500 milliseconds.

Suppose that we know that the last I/O request accessed cylinder 3000. (Cylinders are numbered sequentially: $1, 2, \ldots, 8192$.)

a. What is the expected (average) number of cylinders that will be traveled due to the very next I/O request to this disk?

b. What is the expected block access time for the next I/O, again given that the head is on cylinder 3000 initially?