Matthew Acs
COP 3540 Assignment #1
7/29/2021

Question 1 (8 points): Consider the B+ tree index shown in Figure 1. 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. Show every step of the insertion and explain the process.

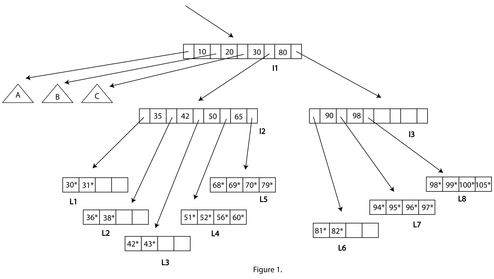
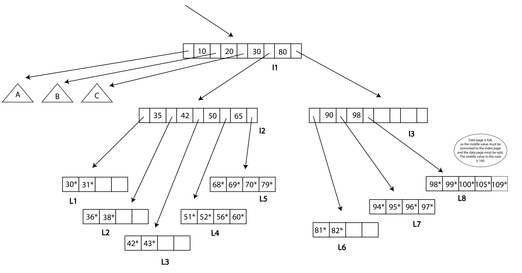
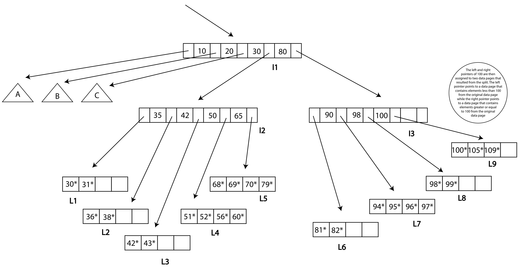


Figure 1.

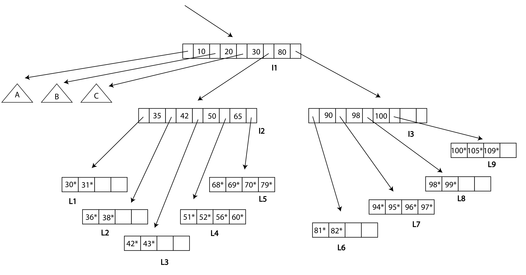
Part 1: Insert a record with search key 109 into the tree.

The first thing that we need to do in order to insert 109 is to compare it to the values in the first index page. Since 109 is larger than 80, we follow the right pointer to the rightmost index page labeled I3. We repeat this process and find that 109 is larger than 98, leading us to follow the pointer to the data page L8. At this point 109 should be inserted into the data page, however the data page is full. Therefore, we need to split the data page by promoting the middle value to the index page I3 and making the left pointer of the middle value lead to a data page containing the elements smaller than the middle value and its right pointer lead to a data page containing the elements greater than or equal to the middle value. This can be done because the index page I3 is not full and promotion is thus possible.



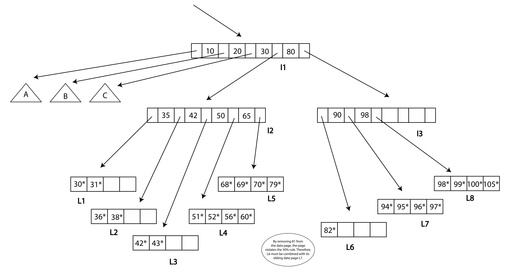


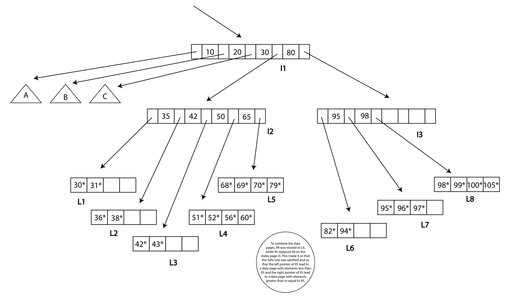
Final Tree for Part 1:



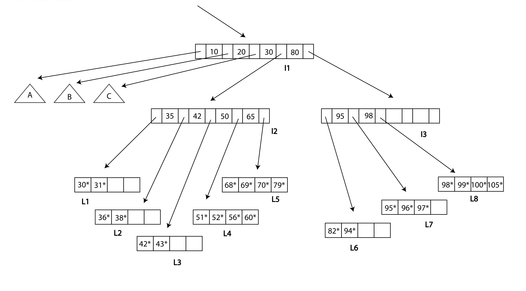
Part 2: Delete the record with search key 81 from the original tree.

The first thing that we need to do in order to delete 81 is to locate it in the tree. Starting at the first index page, 81 is greater than 80, so we follow the rightmost pointer to I3. 81 is then less than 90, so we follow the leftmost pointer of I3 to the data page L6. The data page contains two elements, 81 and 82. By removing 81 from the data page, the page would violate the 50% rule. This means that the data page must be merged with its sibling and the index page must be changed to reflect that merge. In this case, L6 must be merged with L7 by moving 94 to L6 and replacing 90 with 95 on index page I3.





Final Tree for Part 2:



Matthew Acs
COP 3540 Assignment #1
7/29/2021

Question 2 (7 points): Consider the Extendible Hashing index shown in Figure 2. Show the index after inserting an entry with hash value 68.

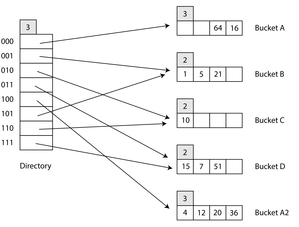


Figure 2.

The first thing that we need to do in order to insert 68 is to find its binary representation. The binary representation of 68 is 1000100, which maps to bucket A2 because the global depth is 3 and the right-most three digits are 100. Bucket A2 is currently full, so we need to increase the local depth of the bucket to 4. Since this depth is larger than the global depth, the global depth also needs to be increased and the directory size must be doubled. Bucket A2 must now be split into two buckets A2 and A3 with local depths of 4 and the contents of the old bucket A2 must be split among these two new buckets. The directory pointers must then be filled in by pointing to the correct buckets.

68 = 1000100

Final Index: