EECS 484 Homework #4

(66 points)

Due: Friday, Nov 8th, 2024 at 11:45 pm (ET)

Please read the following instructions before starting the homework:

This homework must be completed individually and can be submitted on Gradescope. Use entry code **XG8VVB** to self-enroll if you don't have access to the Gradescope course page.

No late days for homework! If you miss the due date, you get 0 points. If your PDF gets modified after the due date, you get 0 points. No exceptions on this.

Honor Code

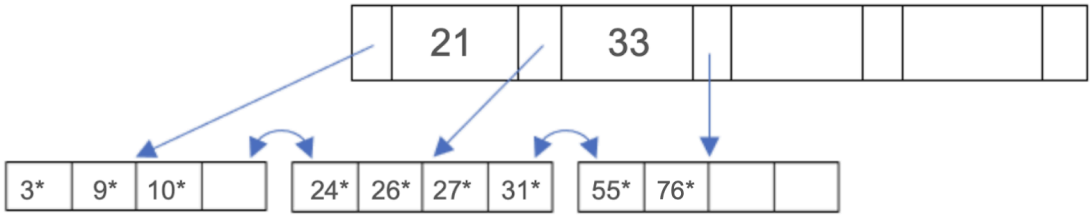
By submitting this homework, you are agreeing to abide by the Honor Code:
I have neither given nor received unauthorized aid on this assignment, nor have I concealed any violations of the Honor Code.

Question 1: B+ Tree Operations (26 points)

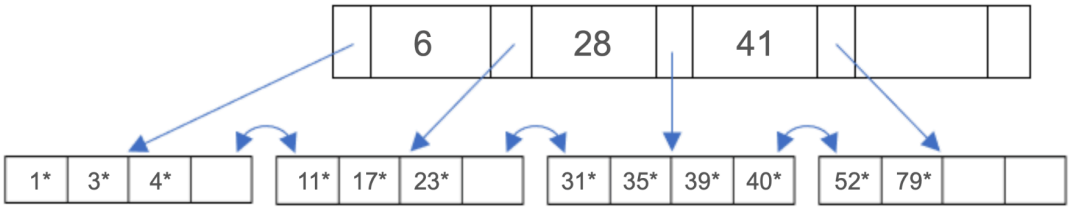
Assume these rules for B+ tree operations in this question:

- The B+ tree is a 5-way search tree.
- The left pointer points to values that are strictly less than (<) the key value.
- During redistribution, shift elements to/from the right node over the left when both are possible redistribution partners.
- During the splitting of a node, the right node will have one more value than the left node.
- For insertions, favor redistribution over splitting.
- Only redistribute one entry at a time.

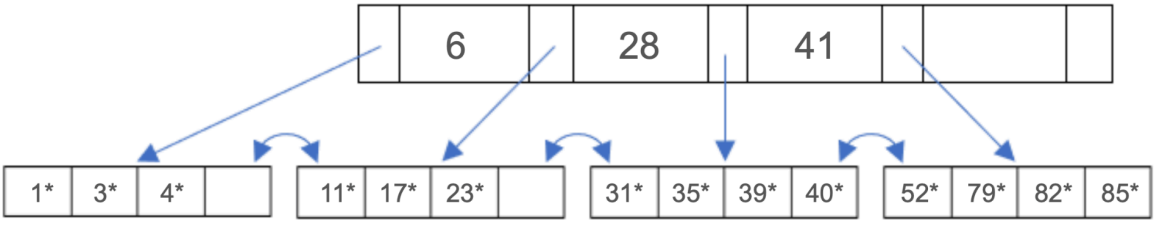
1. Given the tree below, draw the tree after inserting 21* (2 points)



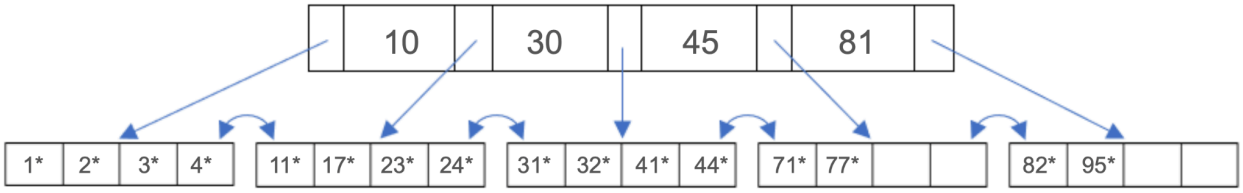
2. Given the tree below, draw the tree after inserting 10*, 25* (4 points)



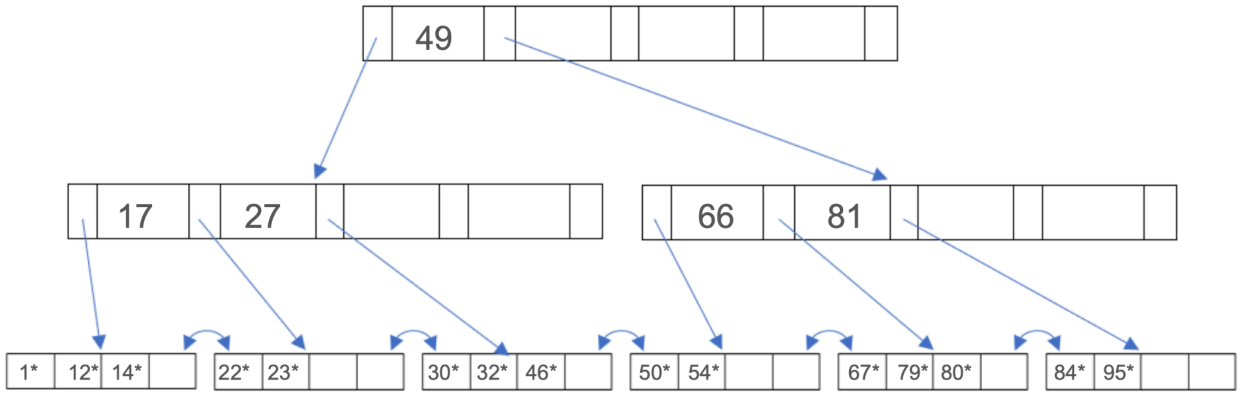
3. Given the tree below, draw the tree after inserting 24*, 32* (4 points)



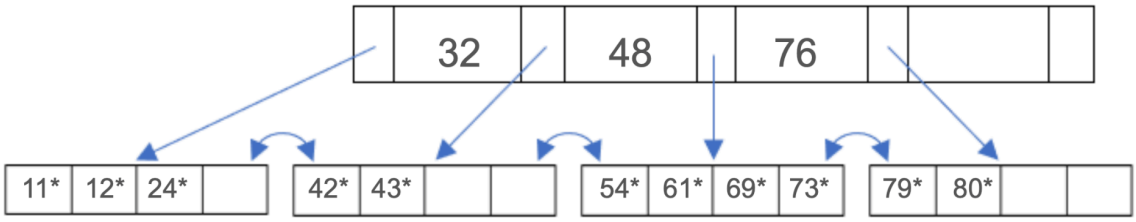
4. Given the tree below, draw the tree after inserting 25* (6 points)



5. Given the tree below, draw the tree after deleting 50*, 67* (6 points)

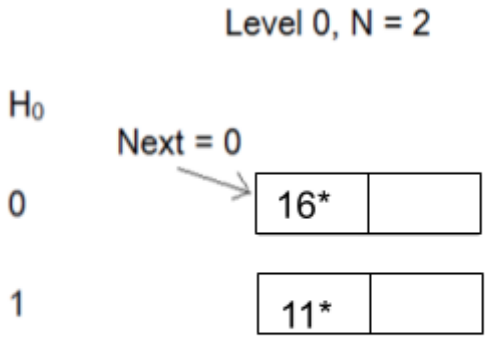


6. Given the tree below, draw the tree after deleting 42* (4 points)



Question 2: Linear Hashing (22 points)

For Question 2, assume the split policy used is splitting when inserting into a full bucket. Assume we initialize the Linear Hashing index as follows:



The hash function that will be used:

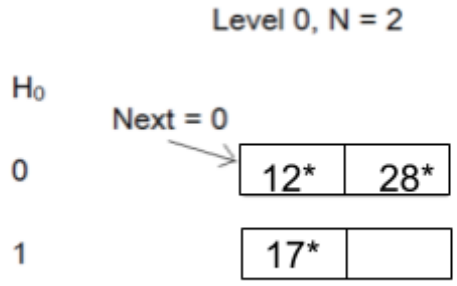
$$h_i(value) = value \bmod (2^i N)$$

Assume that we use the same algorithm discussed in the lecture. N is the initial number of buckets when we first create the hash index, which does not change as the index grows. i indicates the level of the hash index, where every time we split the bucket at the end of the hash index, we increment the level by 1. h_{i+1} will be used for all the buckets above the split pointer, and h_i will be used for all the buckets at or below the split pointer.

1. What is the minimum number of insertions needed such that the index structure has 4 buckets and is at Level = 1? Give an example insertion sequence for your answer. (4 points)

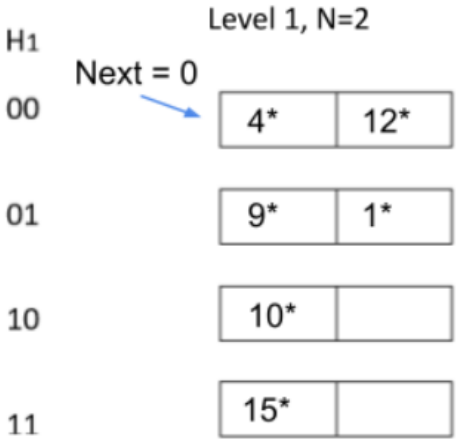
2. For each subquestion, draw what the given index structure looks like after the listed insertion. Remember to show the "next" pointer and level post-insertion. (18 points)

2.1. Insert 14 = $(1110)_2$ (4 points)

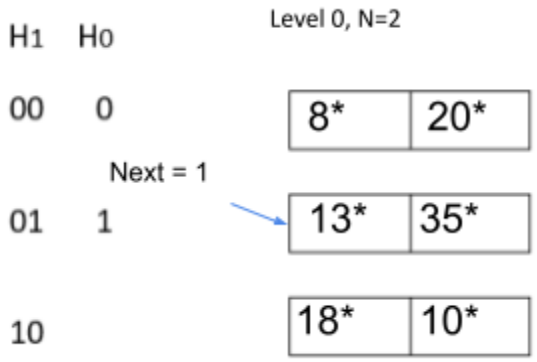


2.2. Insert 31 = $(11111)_2$ (4 points)

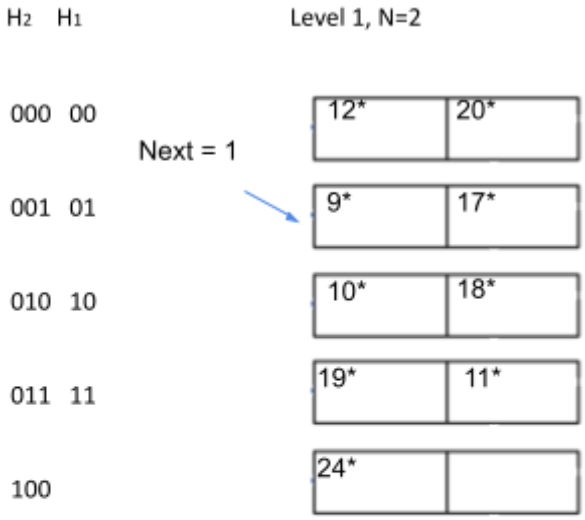
2.3. Insert 19 = $(10011)_2$ (2 points)



2.4. Insert 14 = $(1110)_2$ (4 points)



2.5. Insert 21 = $(10101)_2$ (4 points)



Question 3: Sorting (18 points)

Consider sorting two datasets on a modern desktop machine with small memory. Assume that we have a 484 GB dataset and a 376 GB dataset.

For this question:
1 GB = 1024 MB, 1 MB = 1024 KB, 1 KB = 1024 B
Page size: 16 KB
Show your work for **BOTH** datasets to receive full credit.

3.1) Assume the DBMS has 1 MB of memory (RAM) to use for its buffer pool. How many passes will be required to sort both datasets using general external sort? (4 points)

3.2) Assume the DBMS has 1 MB of memory (RAM) to use for the buffer pool. What is the total I/O cost to sort both datasets using general external sort (in terms of GB)? (4 points)

3.3) What is the minimum amount of memory (RAM) to use for the buffer pool such that the DBMS can sort **BOTH** datasets in 4 passes or less (in terms of MB)? (6 points)

3.4) Assume the DBMS has 1 MB of memory (RAM) to use for the buffer pool. What is the largest dataset (to the nearest GB) that can be sorted using general external sort using 3 passes? (4 points)