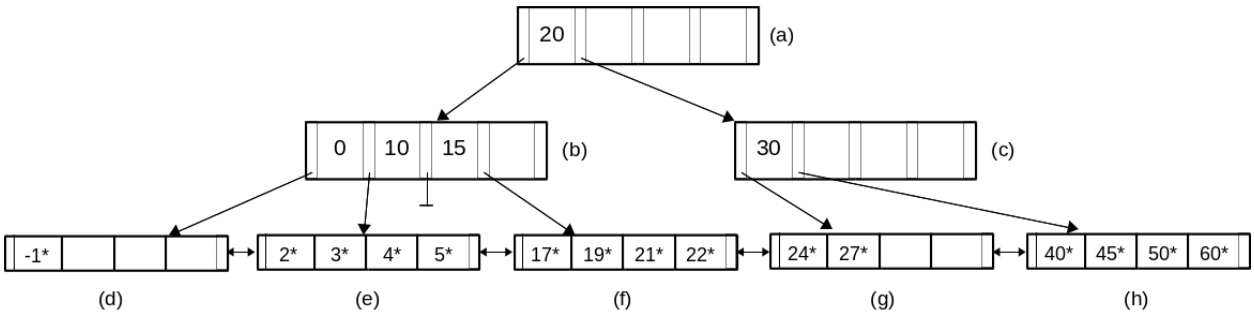
Lab 6: Indexing, Sorting, and Advanced SQL

You may work with one other person on this lab. To submit your assignment, place a PDF in your `~/cs44/labs/6/` *directory and use* `handin44` *to electronically submit the lab. Be sure both names are on the document and that you select the partner option. Your solution is due 11:59pm on* **Sunday, April 19, 2015**.

B+ Tree Indexes

1. Consider the B+ tree in the following figure of order $d = 2$. List all violations of a property of B+ trees. For each violation, simply list the violation and the offending node. For example, if we had a hypothetical node (x) that had too many values, you would list "(x) exceeds $2d$ keys". Note that not all nodes are in violation, and some nodes may have multiple violations.



2. Using **Figure 10.27** of Ramakrishnan and Gehrke
   (a) show the resulting tree after the key **3** has been inserted. Only redraw the relevant portions of the tree (i.e., the modified subtree).
   (b) How many page reads and page writes were required to complete the insertion?
3. **Exercise 10.2, subproblems 1,3-5**, Ramakrishnan and Gehrke. For 3, only draw the relevant portions of the tree. Use the algorithms covered in class (splitting for inserts, redistribution and merge for deletes). The labels for the nodes begin with an `I` for all non-leaf nodes and an `L` for all leaf nodes.
4. Briefly, explain how bulk loading a B+ Tree improves search performance relative to individually inserting each item from a relation (as you did in Lab 5)

Hash Indexes

5. **Exercise 11.1, subproblems 4-7**, Ramakrishnan and Gehrke.
6. **Exercise 11.7**, Ramakrishnan and Gehrke.
7. For the following question, assume one block can fit either a) K key-value pairs plus 1 pointer or b) P pointers to other pages (i.e., `PageId`s). Be brief in your responses:
   (a) For static hashing, what is the minimum number of pages needed to represent a hash table with M buckets? How about for extendible hashing?

(b) For a global depth of G, what is the range of possible values for the *local depth* of any page containing entries? Is there a limitation (minimum or maximum) on the number of pages where the local depth equals the global depth?

(c) What is one key advantage of linear hashing over extendible hashing?

Sorting

8. Assume you have a file with 2,000,000 pages and 17 available buffer pages. Answer the following questions. *Note that the equations given in class were approximations; you should read Section 13.3 for the exact formulas for the number of passes and the number of sets in a run.*

   (a) How many runs does Pass 0 produce?

   (b) How many passes will it take to sort the entire file completely?

   (c) What is the total I/O cost of sorting the entire file?

   (d) How many buffer pages are needed to sort the entire file with only one merge pass (i.e., two passes in total)?

9. In class, we discussed an extension to external merge sort called *double buffering*. Consider the possibility of extending this idea to *triple buffering* (i.e., 3 buffer pages per input as opposed to 2). Describe one potential benefit to this approach as well as one key disadvantage.

10. Why might it be beneficial to use a sorted file (with no index) versus an unclustered B+ Tree index for searches? Assume both are indexed on the same attribute.

SQL

11. What is the purpose of the `CHECK` command?

12. For this question, you will define a SQL query to translate the relational operator of *division* using the `EXISTS/NOT EXISTS` commands in lieu of an SQL division operator. Define a query for the following example where you have you have two relations for enrollments:

    Enrollee(studentid, courseid)

    Course(coursid)

    And you want to find the students enrolled in all courses; i.e., *Enrollee*/*Course*