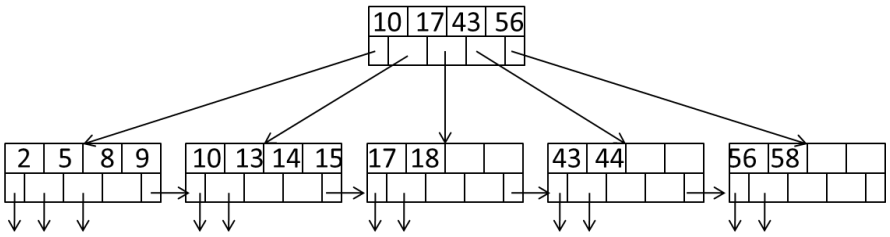
INF 551 – Spring 2016

Homework #4 (100 points)

Due: 11:59pm on 4/15/2016 to Blackboard

1. [30 points] Consider the following B+tree with the degree d = 2, that is, each node (except for root) must have at least two keys and at most 4 keys.



   a. Draw the updated B+tree after first inserting 16.
   b. Draw the updated B+tree after deleting 56 from the tree obtained in part a. Note that when merging two internal nodes, a key from their parent needs to be moved down to the merged node.
   c. Describe the process of finding records in the original tree whose keys are between 11 and 20, such that the number of disk I/O's is minimum. How many blocks I/O's are needed for the process? (Ignore the time to look up the records themselves).

2. [35 points] Consider natural-joining tables R(a, b) and S(b,c). Suppose we have the following scenario.
   i. R is a clustered relation with 100 blocks and 1,000 tuples
   ii. S is a clustered relation with 200 blocks and 2,000 tuples
   iii. S has a clustered index on the join attribute b
   iv. V(S, b) = 5 (recall that V(S,b) is the number of distinct values of b in S)
   v. 12 pages (i.e., page size = block size) available in main memory for the join
   vi. Assume the output of join is given to the next operator in the query execution plan (instead of writing to the disk) and hence the cost of writing the output is ignored.

   Describe the steps (what it does, input, output, their sizes, etc.) in each of the following join methods. What is the total number of block I/O's (read and write of block) needed for each method? Which method has the lowest cost? Which has the highest cost? Compare the two methods and explain the difference in cost.
   a. Nested-loop join with R as the outer relation
   b. Nested-loop join with S as the outer relation
   c. Sort-merge join (assuming sorting uses only 10 pages of the memory and merging uses 11 pages). Note that two types of merging may occur here. First, merging is needed in sorting each relation (so this is intra-relation merging); second, merging is needed in joining two relations (so inter-relation merging) where the method needs to ensure that there is a buffer for each run from both relations. So, the intra-relation merging may be stopped as long as the memory is sufficient to perform the inter-relation merging.
   d. Simple sort-based join (here each relation is completely sorted first, before join)

e. Partitioned-hash join (assuming hashing of R and S only uses 11 pages of the memory, i.e., 10 buckets will be generated.)
f. Index join (ignore the cost of index lookup). Assume that one of the available buffer pages is used for index lookup (holding index node or data block).

3. [35 points] Consider the Iris data set available at: https://archive.ics.uci.edu/ml/datasets/Iris
   a. Draw a scatter plot for the sepal length (1st column of the data set) as x-axis and the petal length (3rd column) as y-axis. Are the two lengths correlated judged from the plot?
   b. Compute the Pearson's correlation coefficient (use population instead of sample co-variance and standard deviations here). What does the coefficient say about their correlation? Is it consistent with the observation from scatter plot?
   c. Build a two-bucket history using the equi-width, equi-depth, and maxDiff method. Compute weighted variance of buckets for each method. What method produces the best histogram according to the variance?