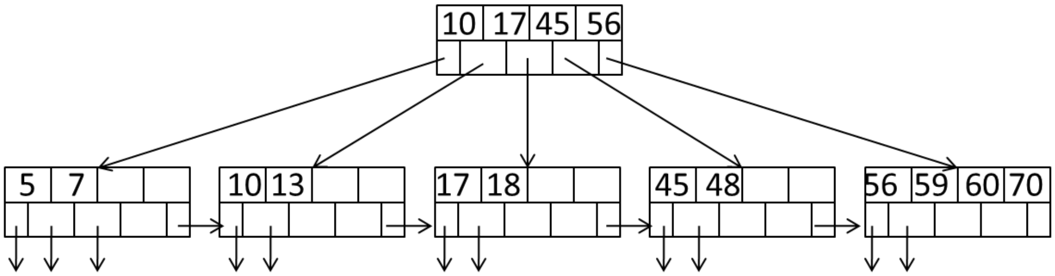
INF 551 – Spring 2019

Homework #4

Due: April 19, Friday
100 points

1. [40 points] Consider the following B+tree for the search key “age. Suppose the degree d of the tree = 2, that is, each node (except for root) must have at least two keys and at most 4 keys.



   a. Describe the process of finding keys for the query condition “age >= 10 and age <= 50”. How many blocks I/O’s are needed for the process?
   b. Draw the updated B+tree after inserting 49, 50, and 51 into the tree. Show the tree after all insertions.
   c. Draw the updated tree after deleting all even ages from the leaf nodes of the tree obtained in part b. Follow this order in deletion: first remove 10, then 18, 48,50, 56, 60, and 70. Show the tree after all deletions. Draw intermediate tree after each insertion or deletion can get partial points if your final answer is incorrect.

2. [60 points] Consider natural-joining tables R(a, b) and S(a,c). Suppose we have the following scenario.
   i. R is a clustered relation with 20,000 blocks and 200,000 tuples
   ii. S is a clustered relation with 8,000 blocks and 80,000 tuples
   iii. S has a clustered index on the join attribute a
   iv. V(S, a) = 100 (recall that V(S, a) is the number of distinct values of a in S)
   v. 102 pages 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 (including input, output at each step, and their sizes) in each of the following join algorithms. What is the total number of block I/O’s needed for each algorithm? Which algorithm is most efficient?

   a. Nested-loop join with R as the outer relation
   b. Nested-loop join with S as the outer relation
   c. Sort-merge join (assume only 100 pages used for sorting and 101 pages for merging). When the number of runs of a relation is too large for merging, the runs will be further merged first. Select the relation with larger number of runs for further merging if both have too many runs.
   d. Simple sort-based join (same assumption as above)

e. Partitioned-hash join (assume 101 pages used in partitioning of relations and no hash table used for lookup in joining buckets)

f. Index join (ignore the cost of index lookup)

Submission:

1. Please conclude your answers with in a pdf file with the naming convention: <Firstname>_<Lastname>_hw4.pdf, all lowercases.
2. For question 1.b and 1.c, you can draw them and take a picture, but make sure your photo is clear enough (Drawing by computer is preferred). For other questions, please type your answers, which should be searchable and copyable, not an image.
3. Your answers should be in the same order as the order of questions. For each question, if you provide multiple answers, we will only grade the first one.