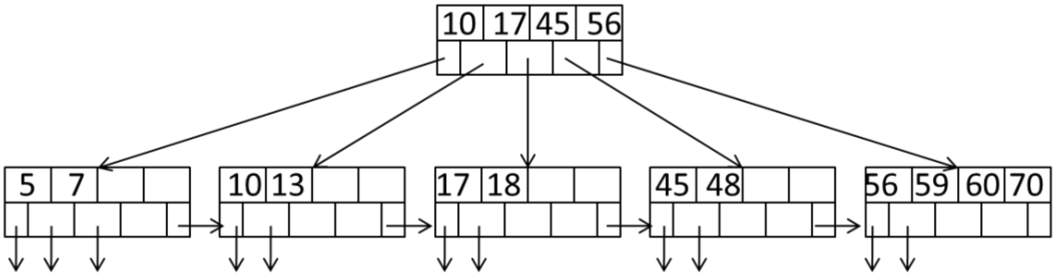
INF 551 – Fall 2018

Homework #4

Due: November 4, Sunday
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 >= 20 and age != 45". How many blocks I/O's are needed for the process?
   b. Draw the updated B+tree after inserting 65 into the tree.
   c. Draw the updated trees after deleting all odd ages from the leaf nodes of the tree obtained in part b. Assume the deletion goes from left to right. That is, first remove 5, then 7, 13, 17, 45, etc. Show the tree after each deletion.

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 12,000 blocks and 120,000 tuples
   ii. S is a clustered relation with 5,000 blocks and 100,000 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. 100 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
   d. Simple sort-based join
   e. Partitioned-hash join
   f. Index join (ignore the cost of index lookup)