CS392 Database System Concept

Assignment 7

Due April 28th, 2014

1. (15')Given key values: (2, 3, 5, 7, 11, 19, 23, 29, 31). Construct B^+-Tree under the following three cases (assume the key is inserted in ascending order)
   a. Four pointer per node
   b. Six pointer per node
   c. Eight pointer per node
2. (15')For each case in problem 1, show the tree after each of the following series of operations:
   a. Insert 9
   b. Insert 10
   c. Insert 8
   d. Delete 23
   e. Delete 19
3. (15')Suppose there is a relation R(A, B, C), with B+ tree index with search key (A, B).
   a. What is the worst case cost of finding records satisfying $10 < A < 50$ using this index, in terms of the number of records retrieved n_1 and the height h of the tree?
   b. What is the worst case cost of finding records satisfying $10 < A < 50$ ^ $5 < B < 10$ using this index, in terms of the number of records n_2 that satisfy this selection, as well as n_1 and h defined above.
   c. Under what condition on n_1 and n_2 would the index be an efficient way of finding records satisfying $10 < A < 50$ ^ $5 < B < 10$.
4. (20')Let relations $r_1(A, B, C)$ and $r_2(C, D, E)$ have the following properties: r_1 has 50,000 tuples, $r_2$has 45,000 tuples, 25 tuples of r_1 fit on one block, and 30 tuples of r_2 fit on one block. Estimate the number of block transfers and seeks required, using each of the following join strategies for $r_1 \bowtie r_2$
   a. Nested-loop join
   b. Block nested-loop join
   c. Merge join
   d. Hash join
5. (15')Suppose that a B+-tree index on building is available on relation department, and that no other index is available. What would be the best way to handle the following selections that involve negation?
   a. $\sigma\neg(\text{building} < \text{“}Watson\text{”})(department)$
   b. $\sigma\neg(\text{building} = \text{“Watson”})(\text{department})$
   c. $\sigma\neg(\text{building} < \text{“}Watson\text{”} \vee \text{budget} < 50000)(department)$

6. (20’)Pipelining is used to avoid writing intermediate results to disk. Suppose you need to sort relation r using sort–merge and merge-join the result with an already sorted relations.
   a. Describe how the output of the sort of r can be pipelined to the merge join without being written back to disk.
   b. The same idea is applicable even if both inputs to the merge-join are the outputs of sort–merge operations. However, the available memory has to be shared between the two merge operations (the merge-join algorithm itself needs very little memory).What is the effect of having to share memory on the cost of each sort–merge operation.