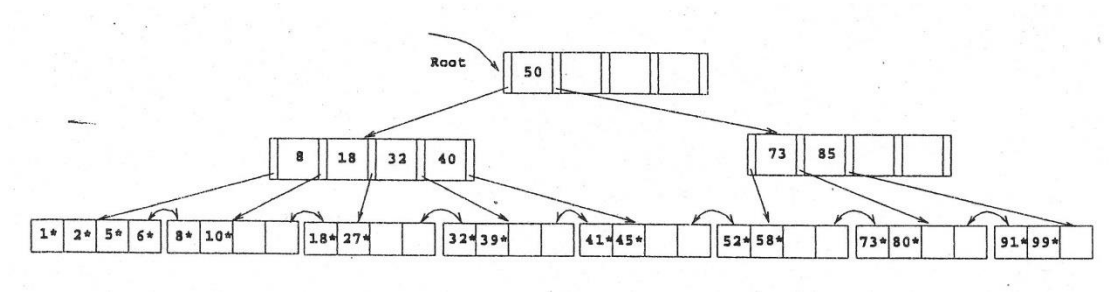
CS 353 Spring 2017
Homework 6
Due: 10 May, Wednesday till 17:00

Q.1 [20 pts] Consider an extendable hash structure where buckets can hold 4 search-key values. Assume that the entries with the key values listed below are inserted using their most significant bits indicated in binary format in parenthesis. Using the extendable hashing method starting with 4 buckets, show the contents of the hash table and the bucket address table after all the integers are inserted. Indicate which insertions lead to bucket splits.

15 (001101), 10 (001010), 40 (101000), 28 (011100), 18 (010010), 20 (010100), 62 (111110), 49 (110001), 52 (110100), 61 (111101), 17 (010001), 25 (011001), 9 (001001), 44 (101100), 34 (100010), 30 (011110), 56 (111000)

Q.2 [24 pts, 8 pts each] Consider the following B+ tree. The maximum number of pointers for each node of the tree is 5 (n = 5). The notation *k** in a leaf node corresponds to a pair of <*p*, *k*>, where *k* is the search key value and *p* is the pointer to the file record with the key value *k*.



(a) Draw the resulting tree after inserting an entry with search key value *k* = 46 and then deleting the entry with seach key value *k* = 52.
(b) Draw the tree after deletion of the entry with seach key value *k* = 91 from the original tree.
(c) Draw the tree after deleting the entries with seach key values *k* = 32, 39, 41, 45, and 73 successively from the original tree.

Q.3 [8 pts, 4 pts each]
(a) Consider a small relation which fits in one block on disk. How many disk block transfers are required to sort this relation?
(b) Consider two small relations each can fit in one block on disk. How many disk block transfers are required to join them, using the block nested-loop algorithm? How many buffer blocks are assumed to be available in main memory for that cost?

Q.4 [18 pts, 6 pts each] Consider the relations R(A, B, C, D, E) and S(A, F, G)
R has 10,000 records, with 50 records fitting on a block. S has 100,000 records, with 100 records fitting on a block. There is no index on any attribute of the relations.

(a) With a memory size of 52 buffering blocks, what will be the cost in terms of the number of block transfers if we use a block-nested loop join to join two relations. Select S as the inner relation of nesting.
(b) Assume we want to join the two relations using a block-nested loop join and limit the cost to 2200 block transfers. What is the smallest number of buffering blocks?
(c) What is the cost of joining two relations using a hash-based join?

Q.5 [30 pts, 6 pts each] Consider a file with 1,000,000 blocks. We want to sort this file using the Sort-Merge algorithm with 120 blocking buffers available in memory.
(a) What will be the initial number of runs?
(b) How many merge passes are required, after generating the initial runs?
(c) Estimate the I/O cost of sorting this file in terms of the number of block transfers.
(d) How many memory buffers are required to sort this file in just one merge pass?
(e) Consider a different file to be sorted. With 120 memory buffers, what is the maximum possible size of the file to sort it in two merge passes (after initial run generation)?