COMP2411 Fall 2023 Class Exercise 7

Student Name: _____

Student ID: _____

Question 1. Consider an unsorted file that has N = 10,000 Movie records. It is stored on a disk with the following configuration:

- Block size (B) = 1000 bytes
- Block pointer size (P) = 6 bytes

A. Suppose a secondary index is constructed on the primary key MovieID (integer, 4 bytes). Calculate the number of block accesses needed for an equality search on the primary key using the secondary index.

B. Suppose that a B+ tree index is constructed on the primary key MovieID (integer, 4 bytes). Calculate the maximum number of block accesses needed for an equality search on the primary key using the B+ tree index.

Question 2. Consider the B+ tree index of order p = 3. Assume the following rule applies for redistributing keys after a leaf node split: Two keys stay in the old node and the remaining keys move to a new node. Answer each of the following questions:

A. Assuming that the tree is initially empty, construct the B+ tree by adding the following set of key values in ascending order.

(2, 3, 5, 7, 11, 17, 19, 23, 29, 31)

B. For the B+ tree derived from the above question, show the updated tree after each of the following operations:

a. Insert 9
b. Insert 10
c. Insert 8
d. Delete 7
e. Delete 5
f. Delete 3

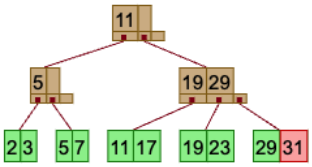
Answers for Question 1: *(floor(X) is the largest integer less than or equal to X; ceiling(X) is the smallest integer greater than or equal to X)*

A. Index entry size (R_i) = MovieID + P = 4+6 = 10 bytes
Index blocking factor (bfr_i) = floor (B/R_i) = floor (1000/10) = 100 entries/block
Number of index entries (N_i) = number of records (N) = 10,000 entries
Number of index blocks (B_i) = ceiling (N_i/bfr_i) = ceiling (10,000/100) = 100 blocks
Number of index block accesses = ceiling ($\log_2 B_i$) = ceiling ($\log_2 100$) = 7 blocks
Total number of block accesses = 7 + 1 = 8 blocks

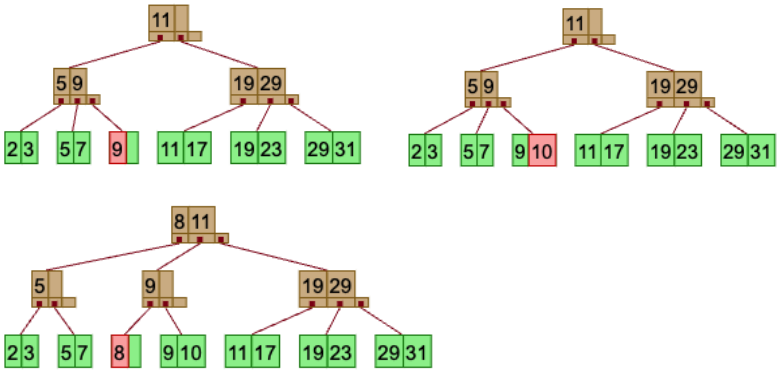
B. *The nodes in a B+ tree are at least half full. Hence, the B+ tree has the maximum height when all its nodes are half full.*
Index blocking factor (bfr_i) = floor ($B*50\%/R_i$) = floor (1000*50%/10) = 50 entries/block
Tree height = ceiling ($\log_{bfri} N_i$) = ceiling ($\log_{50}$ 10,000) = 3
Total number of block accesses = tree height + 1 = 3 + 1 = 4 blocks

Answers for Question 2: *(from: https://goneill.co.nz/btree-demo.php)*

A. Constructed B+ tree using key values (2, 3, 5, 7, 11, 17, 19, 23, 29, 31):



B. After inserting 9, 10, 8, respectively:



After deleting 7, 5, 3, respectively: