CAS CS 460: Introduction to Database Systems – Fall 2020 – Written Assignment #4

Due: 10/25 11:59 PM in Gradescope

Problem 1 [50 pts]

Consider a hard disk drive that has 5 double-sided platters, each surface has 1000 tracks, each track has 256 sectors of size 512 bytes. Each block (disk page) comprises of 8 sectors. The seek time between adjacent tracks in 1ms and the average seek time between two random tracks is 25ms. The disk rotates at a speed of 7200 rpm (revolutions per minute).

Let's say, we have a file of size 1 MB and it contains 2048 equal-sized records.

1. What is the size of a block? How many records fit in a block? How many blocks are required to store the entire file?

2. What is the capacity of each cylinder?

3. What is maximum time (worst case) to read two blocks from the disk (the blocks to be read are part of the same read request and no external factors affect the read latency)?

4. If the file is stored "sequentially", how long will it take to read the whole file? Assume that for sequential writes data are written in adjacent tracks once a track is full.

5. If the blocks in the file are spread "randomly" across the disk, how long will it take to read the whole file?

Problem 2 [50 pts]
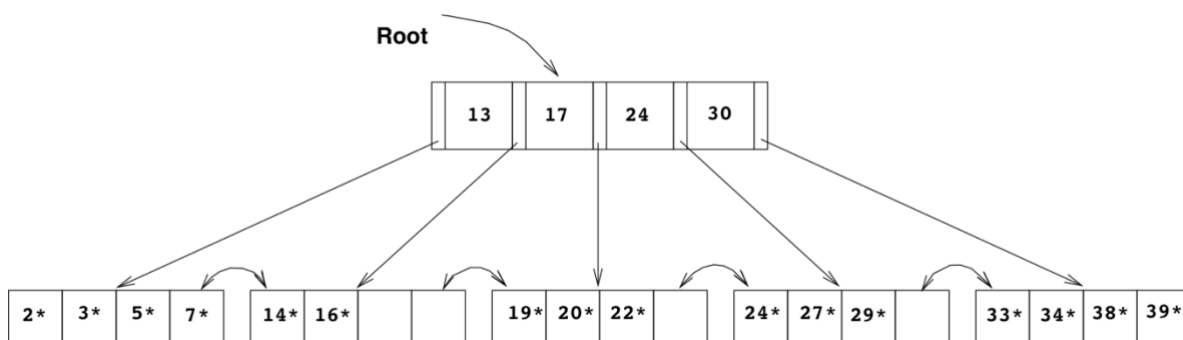
1. Based on the given B+ tree, Identify a list of five data entries such that:

(a) Inserting the entries in the order shown and then deleting them in the opposite order (e.g., insert a, insert b, insert c, delete c, delete b, delete a) results in the original tree.

(b) Inserting the entries in the order shown and then deleting them in the opposite order (e.g., insert a, insert b, insert c, delete c, delete b, delete a) results in a different tree.

2. What is the minimum number of insertions of data entries with distinct keys that will cause the height of the (original) tree to change from its current value (of 1) to 3?

3. Would the minimum number of insertions that will cause the original tree to increase to height 3 change if you were allowed to insert duplicates (multiple data entries with the same key), assuming that overflow pages are not used for handling duplicates?