CAS CS 460: Introduction to Database Systems – Fall 2021 – Bonus Written Assignment #5b

Due: November 17th, 11:59pm in Gradescope

Question 1: External Sorting. Answer the following questions for each of these scenarios, assuming that the generalized external sorting algorithm is used:

(a) A file with 10,000 pages and three available buffer pages.

(b) A file with 20,000 pages and five available buffer pages.

(c) A file with 2,000,000 pages and 17 available buffer pages.

1. How many runs will you produce in the first pass?
2. How many passes will it take to sort the file completely?
3. What is the total I/O cost of sorting the file?
4. How many buffer pages do you need to sort the file completely in just two passes? How would the answer to this question change if you can use heapsort in pass 0?

Question 2: Indexing with a Hash Index.
Suppose that we are using extensible hashing on a file that contains records with the following search-key values:
(449, 124, 654, 831, 1016, 176, 285, 468, 615, 340, 331, 135, 667, 818, 117, 429)

Suppose we load these values into a file in the given order using extensible hashing. Assume that every block (bucket) of the hash index can store up to four (4) values.

Show the structure of the hash index after every 4 insertions, and the global and local depths. Use the hash function: h(K) = K mod 128 and then apply the extensible hashing technique. Using this function, every number is mapped first to a number between 0 and 127 and then we take its binary representation. Then, the extensible hashing technique is applied on the binary representation. Furthermore, initially, you start with a single bucket and a single pointer and the global and local depths are zero (0).

Question 3: Indexing with LSM.
Suppose we want to store 2^{72} entries and that a disk page fits 2^8 entries.

1. How many I/Os would a point query require? (Hint: assume a sorted file on disk)
2. Suppose we want to speed up our queries and we decide to build a B^+ tree index. Compare the cost answering a point query with both approaches only considering the lookup cost, without calculating the cost to build the B^+ tree.
3. Now assume that we use an LSM-tree to store all our entries. What is the cost of a point lookup if we employ an LSM-tree with size ratio 8 and merging policy tiering?
4. How does that cost change if we change the merging policy to leveling?