Assignment 1

Programming language: Java/Python

1) Given a large text file with strings (one string per line), generate a secondary B+ index file on the text file. You need to do the following: Read the contents of the text file and build an in-memory B+ tree index on it. The leaf nodes contain the file offsets into the input file. The non-leaf nodes contain in-memory pointers to other B+ tree nodes. Further, assume that each key (string) is 25 characters long. If the input is shorter than 25 characters, pad it with spaces to make it 25 characters long. If it is longer, truncate it to 25 characters. You can assume that the entire B+ tree index can be loaded in memory for this exercise.

2) Assuming that the B+ tree is built successfully in memory, provide a method to write the contents of this B+ tree index into a binary file. This is usually called the B+ tree index file. Think of how to handle pointers when you write a data structure to disk.

3) Provide a method to lookup the contents of the data file (original file) using the B+ tree index. The method should traverse the B+ tree index, pick up the offset of the input string and read the line prefixed by the same string from the data file. Assume that the keys are unique. (In a more realistic scenario, the data file will contain more than one line prefixed by a specific key. We are keeping it simple for this exercise.)

4) Once the B+ tree index is built, we can perform update and delete operations on the data file using the index! Given a key, implement a method to delete the key from the index. The method will traverse the B+ tree index and get the offset of the key in the data file. Values will not be actually deleted from the data file. The traversal is to ensure that the key exists in the data file. Once a key is found in leaf node of the B+ tree, delete the key from the B+ tree index. Do not bother deleting it from the data file.

5) Provide a method to display the contents of the B+ tree index.