Data Structure and Algorithms Lab

Lab Program 5

1. Design and implement a menu-driven Java program to perform operations on a multi-way search tree(B-Trees) suitable for indexing large amounts of data. The tree should allow multiple keys to be stored in a single node and should maintain balance automatically during insertions and deletions by redistributing or merging keys when required. The program should support inserting new keys, deleting existing keys, and searching for a given key efficiently. It should also provide options to traverse and display the contents of the tree in sorted order and in level-wise structure to visualize the hierarchy. The implementation must ensure that the tree remains balanced after every update so that search, insert, and delete operations can always be performed in logarithmic time. Proper messages should be displayed for unsuccessful search or delete operations, and the program should be tested with a set of sample keys to demonstrate how the structure grows and shrinks dynamically.

2. Design and implement a menu-driven Java program to construct and manipulate a binary heap for a dynamic set of elements. The program should allow the user to insert new elements while maintaining the heap property, delete elements such as the root (maximum in a max-heap or minimum in a min-heap), and efficiently retrieve the highest or lowest priority element depending on the chosen heap type. It should also provide functionality to build a heap from an unsorted array, display the elements in heap order, and perform a heap sort to arrange the data in ascending or descending order.