Programming Assignment:

Objectives

* To use External Merge Sort for sorting the data
* To implement Bulk Loading for building the B+ Tree in two ways:
  + Top-down Approach
  + Bottom-Up Approach
* To compare the performance of both approaches and compare them with insertion without sorting
* Analyze the quality of indices created from each of the algorithms (height, num of nodes)

**Bulk Loading Approaches:**

Bulk Loading: Bulk loading is implemented in 2 ways:

1. Top down insertion: The data is sorted using External MergeSort, and inserted in a top down manner into the index. In this approach, the nodes are always half full, as they get filled in the sorted order and split, and none of the new entries go to the old nodes. But this approach is fast in terms of creation time.
2. Bottom up building: First the leaf node layer is created by populating nodes completely and connecting till all the entries are finished. Then a recursive function builds the internal node. It starts from a node which is designated as the root, and then recursively builds the sub-trees corresponding to all it’s children. The height at a point in the recursion is maintained. The base case is reached when the recursion reaches the node just above the leaf node level. There, the node is populated, and the pointed to the next leaf node is passed above. Then the recursion builds the rest of the internal nodes in the similar manner, while handling the keys and the pointers to the leaf nodes as the recursion passes up. This approach is very efficient in terms of space efficiency of the B+ tree, as almost all the nodes are completely filled. Hence, the height is smaller, and the accesses are faster.