This lab consisted of implementations of a B-tree operations. For this lab I was able to compute the height of the tree, extract the items in the B-tree into a sorted list, return the minimum element in the tree given a depth d, return the maximum element in the tree given a depth d, return the number of nodes in the tree at a given depth d, and given key k, return the depth at which is found on the tree.

The height has a recurrence equation T(n) = 1T(n-1) which will have a runtime of O(n). Extracting the elements of the B-tree and putting them in a list has a recurrence equation of T(n) = 2T(n-1)+ 2n which gives a runtime of O(2^n). Returning the max or min element from a given depth are similar and as such have a recurrence equation of T(n) = T(n-1) which will give a runtime of O(n). The function for returning the number of nodes from a given depth has a recurrence equation of T(n) = 2T(n-1)+n which will have a running time of O(n^2). At last, the function to find the depth of a given depth has a recurrence equation of T(n) = 2T(n-1) + 2n which will also give a running time of O(2^n).

Majority of the operations that I was able to complete have a slow runtime and struggled to complete. I did manage to traverse the B-tree but to traverse it only at a given depth was difficult, nonetheless. All had recursion involved to be able to complete which is common for trees.

Appendix:

Height function lines 80 to 83

Tree to list function lines 127 to 136

Number of nodes at a given depth function lines 138 to 147

Minimum at given depth function lines 149 to 156

Maximum at given depth function lines 158 to 165

Given key located at depth function lines 167 to 179

Code that prints can calls functions lines 195 to 201