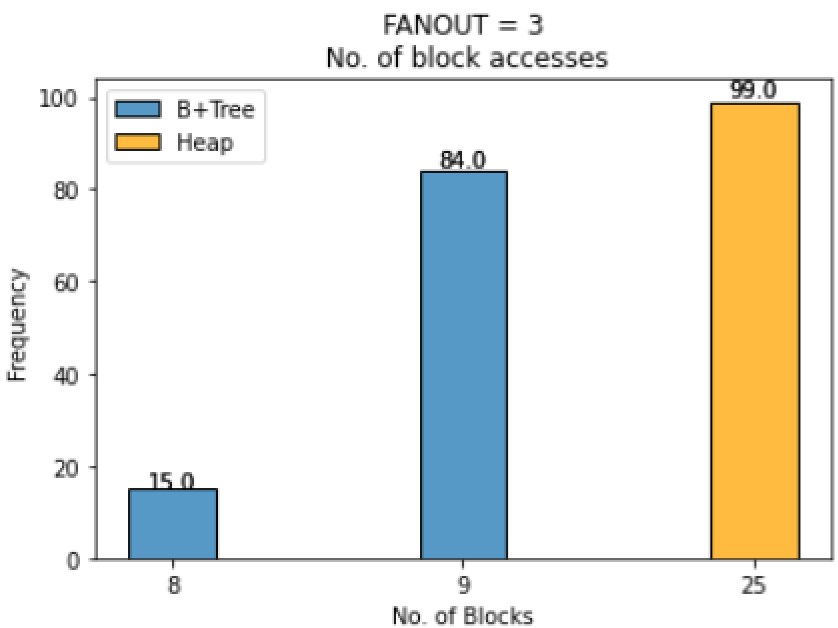
Project Phase 1: B+ Tree

By: Mihir Bani (2019113003), Samay Kothari (2019113017)

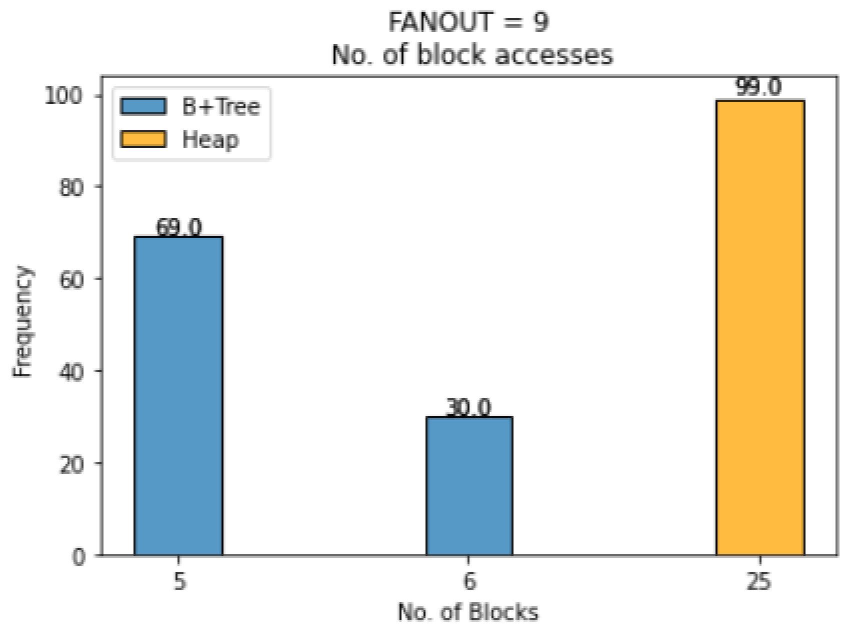
Analysis

To make an analysis of the B+ tree implementation and its efficiency, we insert numbers **1-100** in random order and then run RANGE queries for all keys and count the frequency of the block accesses required in a B+ Tree and a simple unordered heap. This is done for different values of FANOUT and compared.

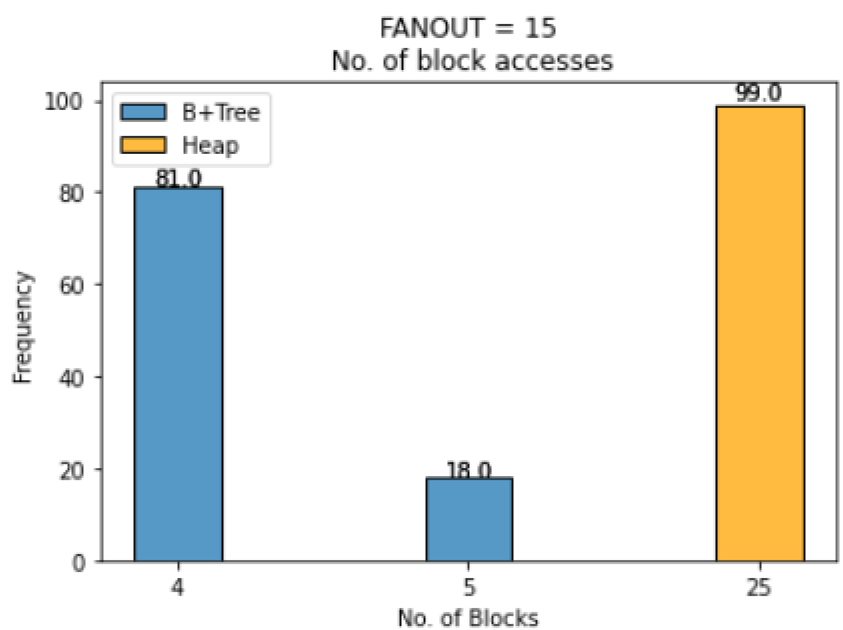
FANOUT = 3



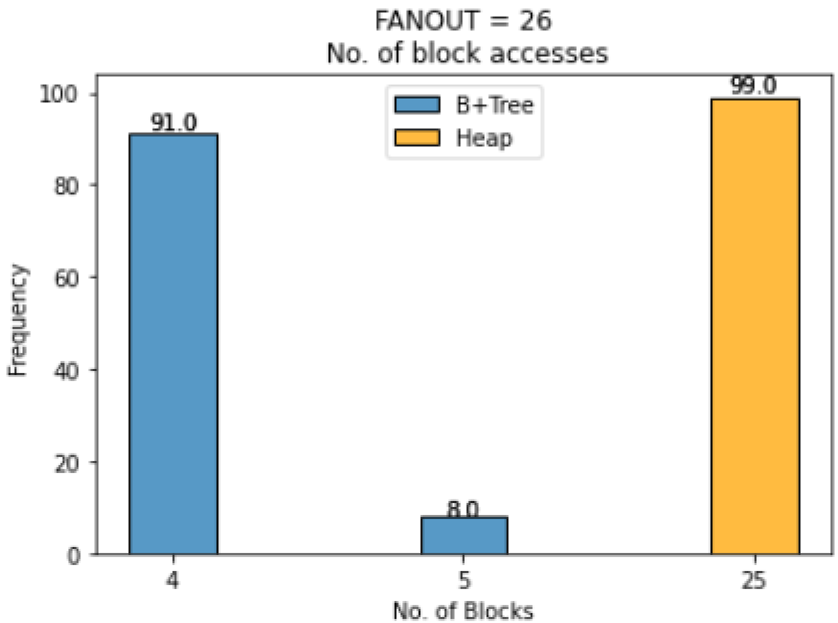
FANOUT = 9



FANOUT = 15



FANOUT = 26



Conclusion

From the respective plots we can see that as the value of FANOUT increases, the number of Block Accesses required using a B+ Tree are very less compared to that of unordered heap showcasing the efficiency and scalability of the B+ indexing.
While the number of block accesses of unordered heap remains constant = 25, the block accesses of the B+ tree are initially high owing to low FANOUT and thus more branches and nodes to remain balanced. As FANOUT increases, the height of the tree decreases and the width increases which lowers the blocks access to find the keys.