3D5- Assignment 3 report  
Task 2 – A Practical Application

Although an output was produced, my bstdb.c code struggled with maintaining a balanced search tree. Numerous attempts were made to employ self-balancing mechanisms like the AVL tree (1) or Red-Black tree (2) approach in my code but unfortunately segmentation faults and errors would continuously arise.

With the bstdb\_stat(void) function I was able to implement tests for my BST in order to produce relevant statistics. One of these tests was a way to measure the average number of nodes visited per search, which turned out to an average of 10.09 nodes judging from my output. While this could say that my BST had reasonable efficiency, the inability to maintain balance indicates a potential performance limitation.

**Task 2 Output:**

Profiling listdb

-------------------------------------------

Total Inserts : 100027

Num Insert Errors : 0

Avg Insert Time : 0.000000 s

Var Insert Time : 0.000000 s

Total Insert Time : 0.048086 s

Total Title Searches : 10002

Num Title Search Errors : 0

Avg Title Search Time : 0.001116 s

Var Title Search Time : 0.006147 s

Total Title Search Time : 11.170246 s

Total Word Count Searches : 10002

Num Word Count Search Errors : 0

Avg Word Count Search Time : 0.001088 s

Var Word Count Search Time : 0.006653 s

Total Word Count Search Time : 10.885648 s

STAT

Avg comparisons per search -> 50024.909368

List size matches expected? -> Y

Profiling bstdb

-------------------------------------------

Total Inserts : 100027

Num Insert Errors : 0

Avg Insert Time : 0.000003 s

Var Insert Time : 0.000028 s

Total Insert Time : 0.318102 s

Total Title Searches : 10002

Num Title Search Errors : 9999

Avg Title Search Time : 0.000000 s

Var Title Search Time : 0.000000 s

Total Title Search Time : 0.006442 s

Total Word Count Searches : 10002

Num Word Count Search Errors : 10001

Avg Word Count Search Time : 0.000000 s

Var Word Count Search Time : 0.000000 s

Total Word Count Search Time : 0.005525 s

=== Binary Search Tree Statistics ===

Total Inserts: 100027

Stored Books: 100027

Average Nodes Visited: 10.09

Tree Height: 40

Press Enter to quit...  
  
**References:**  
1. <https://www.tutorialspoint.com/data_structures_algorithms/avl_tree_algorithm.htm>

2. <https://www.geeksforgeeks.org/introduction-to-red-black-tree/>