**CS 2302 - Data Structures**

**Fall 2019**

**Project 4 - Option B**

**Overview**

In Lab 3, you used binary trees to solve an interesting anagrams problem. In this lab, you are asked to solve exactly the same problem using B-Trees. Run a lot of different experiments to compare the performance of AVL, Red-Black, and B-Trees. For B-Trees, run multiple experiments using different *degrees (max\_num\_of\_keys)* to understand how this parameter affects the performance of your solution. Create tables and plots to report your findings. Justify your selection of experiments and explain why you think those experiments help you compare the different data structures.

**What you need to do**

**Part 1 - Due Thursday, October 24, 2019**

Upload the progress you have made. You need to have at least 50% of the lab done by this date.

**Part 2 - Due Tuesday, October 29, 2019**

Final due date (everything finished - code).

**Part 3- Due Friday, November 1, 2019**

Final due date (report)

**Extra Credit**

Re-do the lab using a B-Tree, run multiple experiments using different values for the degree of the tree, and include in your report how performance changes as the degree varies. Finally, use tables and graphs to compare B-Trees with AVL and Red-Black trees.

**Rubric**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Proficient** | **Neutral** | **Unsatisfactory** |
| **Correctness** | The code compiles, runs, and solves the problem. | The code compiles, runs, but does not solve the problem (partial implementation). | The code does not compile/run, or little progress was made. |
| **Space and Time complexity** | Appropriate for the problem. | Can be greatly improved. | Space and time complexity not analyzed |
| **Problem Decomposition** | Operations are broken down into loosely coupled, highly cohesive methods | Operations are broken down into methods, but they are not loosely coupled/highly cohesive | Most of the logic is inside a couple of big methods |
| **Style** | Variables and methods have meaningful/appropriate names | Only a subset of the variables and methods have meaningful/appropriate names | Few or none of the variables and methods have meaningful/appropriate names |
| **Robustness** | Program handles erroneous or unexpected input gracefully | Program handles some erroneous or unexpected input gracefully | Program does not handle erroneous or unexpected input gracefully |
| **Documentation** | Non-obvious code segments are well documented | Some non-obvious code segments are documented | Few or none non-obvious segments are documented |
| **Report** | Covers all required material in a concise and clear way with proper grammar and spelling. | Covers a subset of the required material in a concise and clear way with proper grammar and spelling. | Does not cover enough material and/or the material is not presented in a concise and clear way with proper grammar and spelling. |