Assignment 4 Report CMPT225

Kale Moskowitz

1. Introduction

Kale Moskowitz
301588993
kom1@sfu.ca
CMPT225 Assignment 4

2. Tree Types

- Binary Search Tree
- Red Black Tree
- Splay Tree

3. Keys Being Inserted

N1 is of size One Million. N2 is of size Ten Million. Each set of data was created by generating a vector of integers 1 to n1 or n2 respectively, then shuffling to vector to ensure random insertion order.

4. Question Being Asked

The goal of the experiment was to compare performance of different search trees under larger scale sets. Specifically how well do BSTs, RBTs, and SPTs perform in terms of

- Insertion Time
- Traversal Time for CountEven()
- Traversal Time to print a visual representation of the Tree.

5. Table of times obtained by program

Insertion Times (ms)

Tree Type	n_1	n_2
BST	480.561	10747.1
RBT	482.997	10251.4
SPT	687.542	16652.2

Traversal Time: `countEven()` (ms)

Tree Type	n_1	n_2
BST	42.721	503.558
RBT	33.425	367.845
SPT	38.679	477.514

Traversal Time: `tdisplay()` (ms)

Tree Type	n_1	n_2
BST	2291.81	24733.5
RBT	2125.52	24216.3
SPT	2980.97	30989.5

6. Short Statement of Observations

- Insertion Times for BST and RBT are quite similar at both scales.
- Splay Trees are much slower during insertion, especially with the larger data size.
- Red-Black Trees are the fastest during traversal for both countEven and tdisplay.

7. Short Statement of Inferences from Data

Red-Black Trees seem to provide the most consistent performance because of their self-balancing nature, whereas splay trees seem to splay more than necessary as seen during insertion.

8. Additional Notes

I think that it would be interesting to test with different organizations of the data set, such as a best case, worst case and average case. I chose to go with randomized data as I assumed this would mirror the average case best.

9. Output from Program

```
Name: Kale Moskowitz
Student Number: 301588993
Login ID: kom1@sfu.ca
Chosen Data Structures:
        Binary Search Tree (BST)
        Red Black Tree (RBT)
        Splay Tree (SPT)
Set Sizes: n1 = 1000000, n2 = 10000000

=== Insertion Times (ms) ===
  BST(n1): 480.561   |  BST(n2): 10747.1
  RBT(n1): 482.997   |  RBT(n2): 10251.4
  SPT(n1): 687.542   |  SPT(n2): 16652.2

=== Time to Traverse (countEven()) ===
  BST(n1): 42.721   |  BST(n2): 503.558
  RBT(n1): 33.425   |  RBT(n2): 367.845
  SPT(n1): 38.679   |  SPT(n2): 477.514

=== Time to Traverse (tdisplay()) ===
  BST(n1): 2291.81   |  BST(n2): 24733.5
  RBT(n1): 2125.52   |  RBT(n2): 24216.3
  SPT(n1): 2980.97   |  SPT(n2): 30989.5
```