Analysis of Algorithms

BLG 335E

Project 3 Report

BEYZA TÜRK 1502220704
turkbe22@itu.edu.tr

Faculty of Computer and Informatics Engineering
Department of Computer Engineering
Date of submission: DATE HERE

1. Implementation

1.1. Data Insertion

After constructing binary search tree (BST) and red-black tree (RBT), I compared the efficiency when storing the given VideoGames.csv data set. As I inserted the data, I updated the cumulative sales for each publisher that is already in the tree. For the red-black tree, I used the color and parent attribute.
According to my total time measurement in microseconds, the Red-Black Tree performed faster than the Binary Search Tree because of its self-balancing structure.

```
Binary Search Tree insertion time:0.251
Red Black Tree insertion time:0.149
```

Figure 1.1: Insertion time

1.2. Search Efficiency

When I performed 50 random searches in an unsorted Binary Search Tree and Red-Black Tree, the recorded search time was 0 ns. My timer precision was not high enough to measure such tiny time intervals, so the result rounded down to 0 ns.When I performed random searches on sorted trees, time increased slightly.

```
Result for 50 random searches unsorted:0
Result for 50 random searches sorted:6e-05
```

Figure 1.2: Search time for RBT

```
Result for 50 random searches unsorted:0
Result for 50 random searches sorted:0.0047
```

Figure 1.3: Search time for BST

1.3. Best-Selling Publishers at the End of Each Decade

I printed a list of the best seller publishers at the end of each decade (1990, 2000, 2010, 2020) for both Binary Search Tree (BST) and Red-Black Tree (RBT). For RBT, pre-order traversal displays node depth and color.I used dashes to indicate the depth of the node as given output sample. Best seller and display of the RBT are as follows.

```
End of the 1990 Year
Best seller in North America: Nintendo - 160.02 million
Best seller in Europe: Nintendo - 30.03 million
Best seller rest of the World: Nintendo - 5.65 million
End of the 2000 Year
Best seller in North America: Nintendo - 334.75 million
Best seller in Europe: Nintendo - 101.97 million
Best seller rest of the World: Nintendo - 15.76 million
End of the 2010 Year
Best seller in North America: Nintendo - 722.26 million
Best seller in Europe: Nintendo - 350.91 million
Best seller rest of the World: Electronic Arts - 89.2 million
End of the 2020 Year
Best seller in North America: Nintendo - 814.43 million
Best seller in Europe: Nintendo - 418.36 million
Best seller rest of the World: Electronic Arts - 126.82 million
```

Figure 1.4: Best Seller Publishers

```
(BLACK) Imagic
-(BLACK) Data Age
--(RED) BMG Interactive Entertainment
---(BLACK) Answer Software
----(BLACK) Activision
-----(RED) 989 Studios
------(BLACK) 3DO
-------(RED) 20th Century Fox Video Games
--------(BLACK) 10TACLE Studios
---------(RED) 1C Company
--------(BLACK) 2D Boy
-------(RED) 5pb
--------(BLACK) 505 Games
---------(RED) 49Games
--------(BLACK) 989 Sports
---------(RED) 7G//AMES
------(BLACK) ASCII Entertainment
-------(BLACK) ASC Games
--------(RED) AQ Interactive
-------(RED) Acclaim Entertainment
--------(BLACK) ASK
---------(RED) ASCII Media Works
---------(RED) Abylight
--------(BLACK) Ackkstudios
---------(RED) Accolade
---------(RED) Acquire
-----(RED) Agetec
------(BLACK) Adeline Software
-------(BLACK) Activision Value
--------(RED) Activision Blizzard
-------(BLACK) Agatsuma Entertainment
```

Figure 1.5: Preorder Print of RBT

1.4. Final Tree Structure

Binary Search Tree becomes unbalanced if sorted data is inserted, it behaves like a linked-list. Height becomes O (n) which causes inefficiency for search and insert operations.However Red-Black tree maintains its balance whether the data sorted or unsorted.Coloring the nodes and rotations keep the height of tree close to O (logn). This balanced sturcture ensures efficient and consistent performance for insertion and search operations.

1.5. Write Your Recommendation

Based on my time measurements for both search and insert operations, I recommend using Red-Black Tree for managing the dataset.Unlike the Binary Search Tree, complexity does not depend on input's order because of RBT's self-balancing structure. While BST can degrade to O(n), RBT maintains consistent O(log n) time complexity.

1.6. Ordered Input Comparison

For unsorted input the search time rounded to 0 ns, due to CPU caching optimizations for both Red-Black Tree and Binary Search Tree. For sorted input, BST search time increased to 0.0047 ns because tree became into linked list structure, it lost its efficieny O(log n) to O(n). Unlike BST, RBT's search time increased slightly to 6e-05 ns. RBT consistently performs O(log n) complexity because of its balanced structure.

Table 1.1: Search Time Comparison Between BST and RBT

Tree Structure	Unsorted Search Time (ns)	Sorted Search Time (ns)
Binary Search Tree (BST)	0	0.0047
Red-Black Tree (RBT)	0	6e-05