Analysis Report

for

CENG-202 PROJECT-1

Mustafa Talha Arslan (140144005)

mustafatalha.arslan@stu.thk.edu.tr

University of Turkish Aeronautical Association

Department of Computer Engineering

17 December 2016

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Mustafa Talha Arslan	25/03/17	Initial work	1

1. Introduction

1.1 Purpose

The purpose of this document is analyzing the test results of Binary Search Tree(as mentioned BST in the report), Binary Heap and Red-Black Tree with respect to three test(Insert 500.000 integers, Search 50.000 integers and Delete 50.000 integers). As a bonus part, AVL and Red-Black Tree were compared.

1.2 Properties of Test Environment

Toshiba Satellite P50-B-116 Notebook with Windows 10 Home Language

2.50 GHz 4th generation Intel® Core™ i7-4710HQ processor

16,384 (8,192 + 8,192) MB DDR3L RAM (1,600 MHz)

IDE: Dev-C++

2. Test Results

	INSERT	SEARCH	DELETE
Binary Search Tree	0.152 s	0.001 s	0.138 s
Heap	0.039 s	N/A*	0.039 s**
Red-Black Tree	0.422 s	0.016 s	1.195 s
AVL Tree**	0.637 s	0.020 s	0.497 s

* **Search for Heap is not included for this project and DeleteMin used for deletion.

***AVL Tree tested for the comparison with Red-Black Tree.

3. Complexity Analysis

3.1 Time Complexities of Algorithms

- **3.1.1 Average Time Complexities**

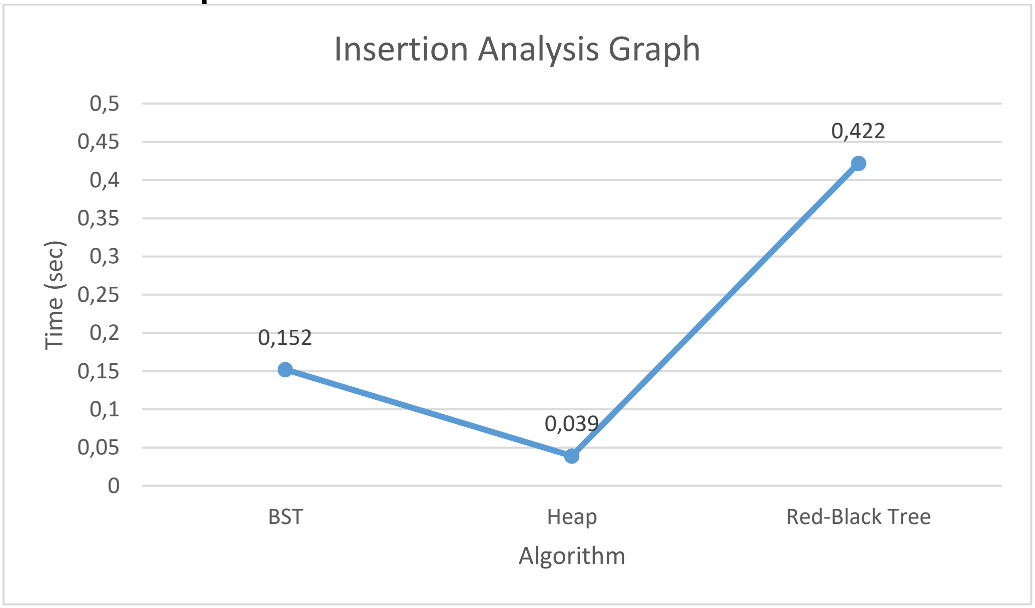
	INSERT	SEARCH	DELETE
Binary Search Tree	O(log n)	O(log n)	O(log n)
Heap	O(1)	O(n)	O(log n)
Red-Black Tree	O(log n)	O(log n)	O(log n)

- **3.1.2 Worst Case Time Complexities**

	INSERT	SEARCH	DELETE
Binary Search Tree	O(n)	O(n)	O(n)
Heap	O(n)	O(log n)	O(log n)
Red-Black Tree	O(log n)	O(log n)	O(log n)

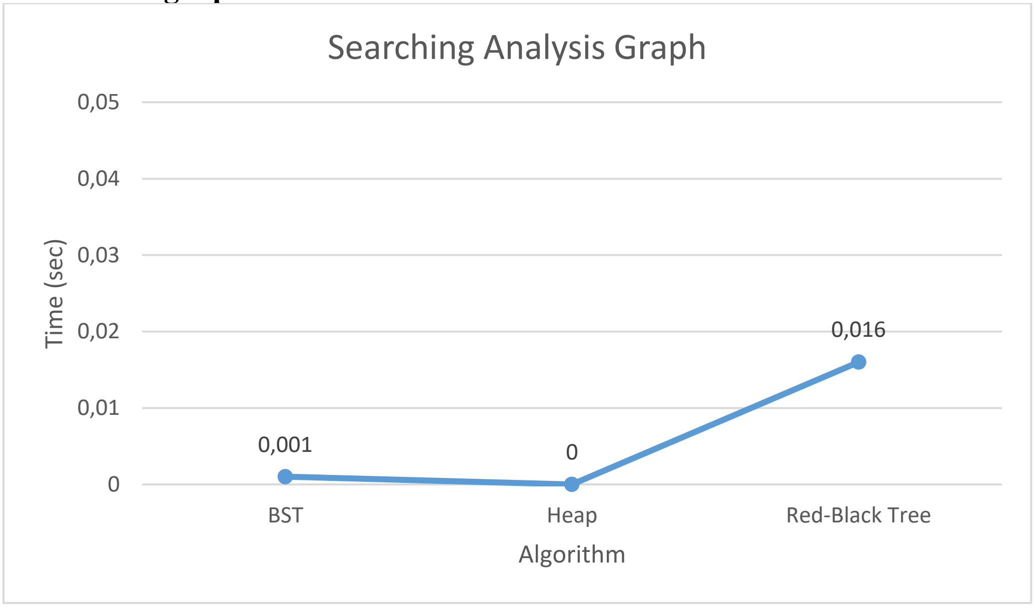
3.2 Analysis

- **3.2.1 Insertion Operation**



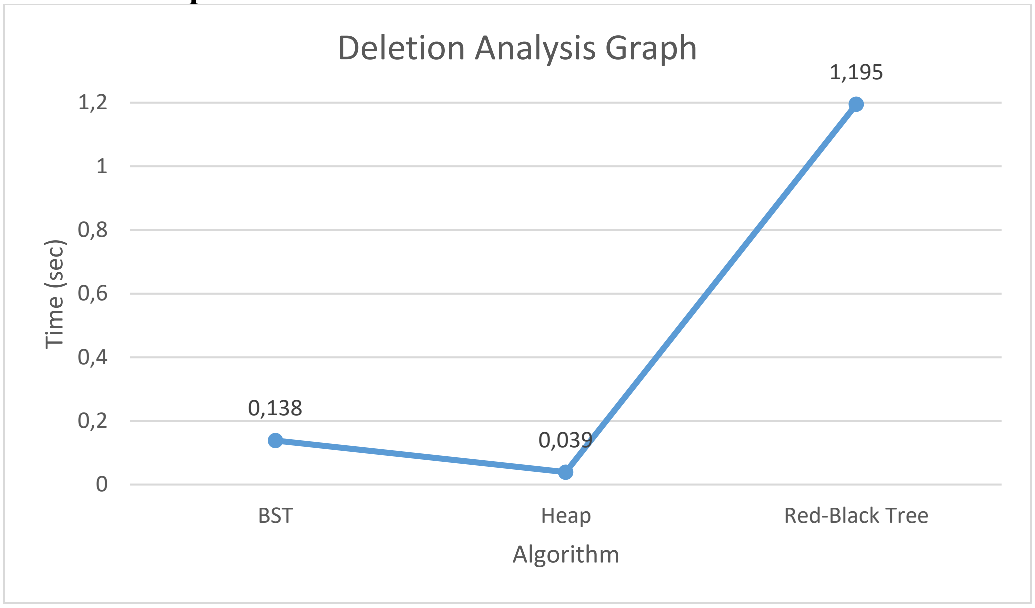
According to time complexities, Binary Search Tree and Red-Black Tree should be slower then Heap for insertion. Then, the test results show that heap has best time for insertion operation, then BST and the slowest algorithm is Red-Black Tree. Even BST and RBT have same complexity, they have different times because of test data.

- 3.2.2 Searching Operation



Searching for Heap is not included for this analysis, so Binary Search Tree and Red-Black should have same or close times. Then, the rest results show that BST and RBT have very close times.

- **3.2.3 Deletion Operation**



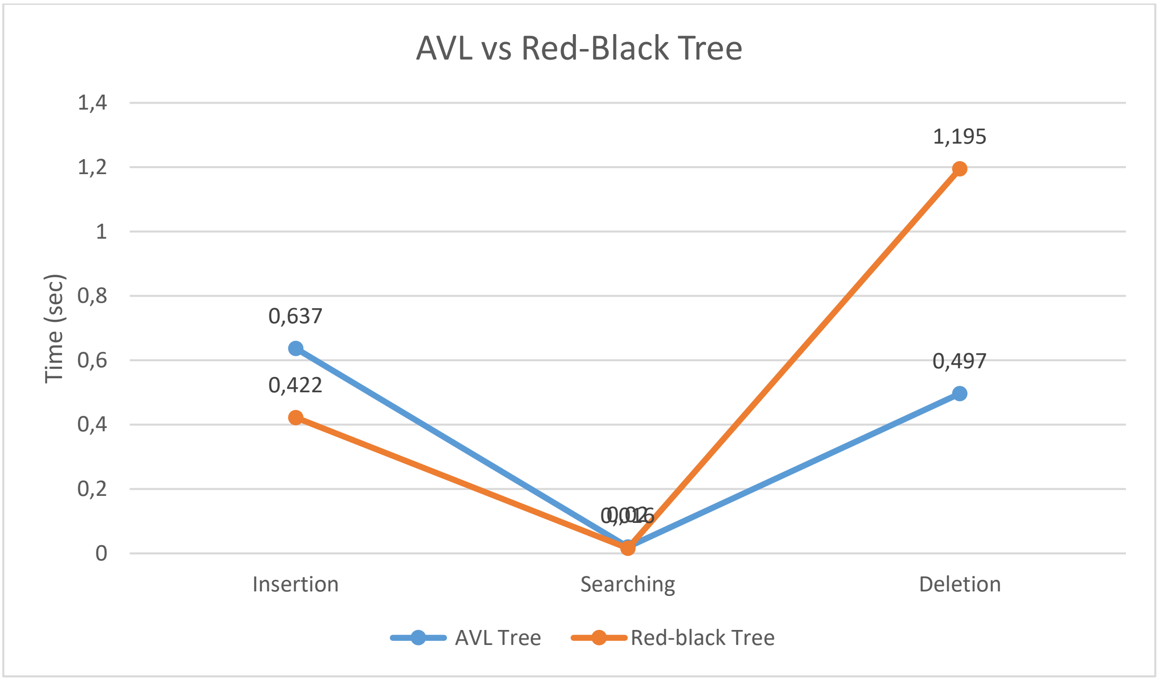
According to time complexities, all algorithms should have same/close times on average conditions for deletion but Delete Minimum used for Heap instead of deletion. Because of this, heap has the shortest time with respect to test results. Binary Search Tree has 2nd shortest time, then the slowest algorithm is Red-Black Tree. Even BST and RBT has same complexity, they have very different test result, so test has some errors.

4. AVL Tree vs Red-Black Tree

	INSERT	SEARCH	DELETE
Red-Black Tree	0.422 s	0.016 s	1.195 s
AVL Tree	0.637 s	0.020 s	0.497 s

AVL Tree and Red-Black Tree was tested with Insertion, Searching and Deletion operations.

	AVERAGE			WORST CASE		
	INSERT	SEARCH	DELETE	INSERT	SEARCH	DELETE
Red-Black Tree	O(log n)	O(log n)	O(log n)	O(log n)	O(log n)	O(log n)
AVL Tree	O(log n)	O(log n)	O(log n)	O(log n)	O(log n)	O(log n)



According to time complexities, AVL Tree and Red-Black Tree should have same/close times for every operation and condition. Then, the test results show that RBT is a little bit shorter than AVL for Insertion and Searching, but they have close times. For Deletion operation, AVL is shorter than RBT, so the test result may have some errors.