**RUN-TIME EFFICIENCY ASSESSMENT FOR THE SELECTION OF OPTIMIZED DATA STRUCTURES FOR JAVA PROGRAMS**

A PROJECT REPORT SUBMITTED BY

R.N.G.J.H. NAWARATHNA

(S/16/417)

to the

**DEPARTMENT OF STATISTICS AND COMPUTER SCIENCE**

*in partial fulfillment of the requirement*

*for the award of the degree of*

**BSc. (Honors) in Computer Science**

of the

**UNIVERSITY OF PERADENIYA**

**SRI LANKA**

**2022**

**DECLARATION**

I do hereby declare that the work reported in this project report was exclusively carried out by me under the supervision of Mr. Prabhath Gunathilake. It describes the results of my own independent work except where due reference has been made in the text. No part of this project report has been submitted earlier or concurrently for the same or any other degree.

Date: ……………………. …………………………………………….

Signature of the Candidate

Certified by:

1. Supervisor: Mr. Prabhath Gunathilake

Date: ……………………. Signature: ………………………………….

1. Head of the Department: Dr. Ruwan Nawarathna

Date: ……………………. Signature: ………………………………….

**RUN-TIME EFFICIENCY ASSESSMENT FOR THE SELECTION OF OPTIMIZED DATA STRUCTURES FOR JAVA PROGRAMS**

**R.N.G.J.H. Nawarathna (S/16/417)**

Department of Statistics and Computer Science, University of Peradeniya,

Peradeniya, Sri Lanka

Data structures help programmers a lot in terms of organizing, storing, and handling data more efficiently. When it comes to Java, the set of built-in optimized data structures has many advantages because they can be used depending on the programmer's needs. These include data structures like Arrays, Lists, Maps, etc. This particular research work intends to make those programmer's needs accessible more efficiently by providing guidance or assistance by comparing and analysing the run-time behavior of different Java data structures with a machine learning approach. We have taken the List, and the Map interface from the Java Collection interface into account to carry out the research. By having the selected data structures benchmarked to get the run-time behaviour in the sense how a data structure in a particular program manages to handle the memory and the run-time,

**AcknowledgmentS**

# 

# Table of contents

DECLARATION ii

ABSTRACT iii

ACKNOWLEDGMENTS iv

TABLE OF CONTENTS v

LIST OF FIGURES vi

LIST OF TABLES vii

LIST OF ABBREVIATATION viii

**CHAPTER 1: INTRODUCTION 1**

1.1 Problem Statements 1

1.2 Objectives 1

1.3 Approach 1

**CHAPTER 2: BACKGROUND 2**

CHAPTER 3: LItERATURE REVIEW 3

CHAPTER 4: METHODOLOGY 4

CHAPTER 5: RESULTS AND DISCUSSION 5

CHAPTER 6: CONCLUSIONS 6

CHAPTER 5: FUTURE WORK 7

**REFERENCES 8**

**APPENDIX A 9**

# list of figures

**Figure 1.1:** Table Caption 1

# 

# LIST of tables

**Table 1.1:** Table Caption 1

# LIST of ABBRIVIARIONS

<<Sort according to the alphabetical order >>

CHAPTER 01

# INTRODUCTION

<<Your Text Here>>

CHAPTER 02

# BACKGROUND

<<This chapter is optional>>

CHAPTER 03

# LITERATURE REVIEW

CHAPTER 04

# METHODOLOGY

CHAPTER 05

# RESULTS AND DISCUSSION

CHAPTER 06

# CONCLUSIONS

CHAPTER 07

# FUTURE WORK

# REFERENCES

# APPENDIX A