

CSC2001F 2020 Assignment 2

Instructions

The goal of this assignment is to compare the AVL Tree with a Binary Search Tree, both implemented in Java, using a real-world application to read in and provide access to Cape Town's load shedding data.

Use the requirements as specified in Assignment 1, but this time compare a BST to an AVL tree. Naturally, the names of applications and files will be different and your results from the experiment should be different.

Parts 1-5

(See Assignment 1)

Report

Write a report (of up to 6 pages) that includes the following:

- What your OO design is: what classes you created and how they interact.
- What the goal of the experiment is and how you executed the experiment.
- What test values you used in the trial runs (Part 2 and Part 4) and what the operations counts were in each case. Only show the first 10 and last 10 lines for the trial run where you invoke *printAllAreas ()*.
- What your final results are (use one or more graphs), showing best, average and worst cases for both applications. Discuss what the results mean.
- A statement of what you included in your application(s) that constitutes creativity - how you went beyond the basic requirements of the assignment.
- Summary statistics from your use of git to demonstrate usage. Print out the first 10 lines and last 10 lines from "git log", with line numbers added. You can use a Unix command such as:

```
git log | (ln=0; while read l; do echo $ln\: $l; ln=$((ln+1)); done) | (head -10; echo ...; tail -10)
```

Dev requirements

As a software developer, you are required to make appropriate use of the following tools:

- git, for source code management
- javadoc, for documentation generation
- make, for automation of compilation and documentation generation

Submission requirements

Submit a .tar.gz compressed archive containing:

- Makefile
- src/
 - all source code

- bin/
 - all class files
- doc/
 - javadoc output
- report.pdf

Your report must be in PDF format. Do not submit the git repository.

Marking Guidelines

Your assignment will be marked by tutors, using the following marking guide.

<i>Artefact</i>	<i>Aspect</i>	<i>Mark</i>
Report	Appropriate design of OOP and data structures	10
Report	Experiment description	10
	Trial test values and outputs (Part 2)	10
	Trial test values and outputs (Part 4)	10
	Results - tables and/or graphs	10
	Discussion of results	10
	Creativity	10
	Git usage log	5
Code	Looks reasonable and no obvious inefficiencies	10
Dev	Documentation - javadoc, junit (edited 21 April)	10
	Makefile - make, docs, and clean targets	5