Data Structures and Algorithms 2, Course Project 2023

Submission Checklist - Very Important *Failure to satisfy these submission requirements may result in non-acceptance of your submission or reduced marks.*	
	Deadline is **strictly** on Friday 26th May 2023 at 23:59.
	Included the **complete** **plagiarism declaration form**.
	Included the **completed** **statement of completion** (template below).
	Report is in PDF - **not Word documents or any other format**; no exceptions.
	Source code is included in the submission. **No links to Dropbox, GitHub, or anything else; no exceptions**.
	Archives are in ZIP format - **no RAR, 7z, or any other format**; no exceptions.
	Uploaded size limit is **100Mb** - the PDF report, source code, and any relevant datasets must fit.
	Your **name** and **student ID** are both on the front page of the report.
	Projects must be submitted only through VLE – **submissions made by email or any other way apart from VLE will not be considered**; no exceptions.
	A draft and final submission area is set up in VLE. Only projects submitted in the final submission area will be graded. **Projects submitted to the draft area will not be considered at all**; no exceptions.
	It is your responsibility to ensure that your upload is complete, valid, and not corrupted. You can reupload the assignment as many times as you wish within the deadline. Double-check your upload. **Corrupted uploads cannot be graded**.
	Plagiarism is a serious offence and will not be tolerated.
	This is **NOT** a group project.

AVL Trees vs Red-Black Trees

- Write a command line program with the following behaviour…

- Let p be a random number between 1000 and 3000.
- Create a set X containing p integers. Each integer must be a random number in the range -3000 and $+3000$. Make sure that there are no duplicates in this set.
- Show the size of the set X…

Set X contains ___ integers.

- Let q be a random number between 500 and 1000.
- Create a second set Y containing q integers. Each integer must be a random number in the range -3000 and $+3000$. Make sure that there are no duplicates in this set.
- Show the size of the set Y…

Set Y contains ___ integers.

- Let r be a random number between 500 and 1000.
- Create a third set Z containing r integers. Each integer must be a random number in the range -3000 and $+3000$. Make sure that there are no duplicates in the set.
- Show the size of the set Z…

Set Z contains ___ integers.

- Note that the sets X, Y, and Z may have elements in common.

- Determine the intersection of X and Y and display its size…

Sets X and Y have ___ values in common.

- Determine the intersection of X and Z and display its size…

Sets X and Z have ___ values in common.

- **Insert all the elements in the set X** into an AVL tree, into a Red-Black tree, and into a binary search tree (a BST with no balancing restrictions which is allowed to degenerate). The AVL and RB trees are binary search trees.
- **The three tree implementations must be your own.**

- After inserting all the values into each tree, you must show the number of rotations performed in total (in the AVL and RB tree, not in the BST), the height of the tree, the number of nodes in the tree, and the number of comparison operations (left/right decisions) made in total…

AVL: ___ tot. rotations req., height is ___, #nodes is ___, #comparisons is ___.
RBT: ___ tot. rotations req., height is ___, #nodes is ___, #comparisons is ___.
BST: height is ___, #nodes is ___, #comparisons is ___.

- **Delete all the elements in the set Y** from each of the three trees.
- After deleting all the values, for each tree, you must show the number of rotations performed in total (in the AVL and RB tree, not in the BST), the height of the tree, the number of nodes in the tree, and the number of comparison operations (left/right decisions) made in total…

AVL: ___ tot. rotations req., height is ___, #nodes is ___, #comparisons is ___.
RBT: ___ tot. rotations req., height is ___, #nodes is ___, #comparisons is ___.
BST: height is ___, #nodes is ___, #comparisons is ___.

- **Search for every element in the set Z** in each of the three trees. Note that a search may be successful or not.

AVL: ___ total comparisons required, ___ numbers found, ___ numbers not found.
RBT: ___ total comparisons required, ___ numbers found, ___ numbers not found.
BST: ___ total comparisons required, ___ numbers found, ___ numbers not found.

- In your report, write a high-quality discussion comparing AVL trees, Red-Black trees, and unbalanced BSTs. In this section, make sure to discuss the suitability (or otherwise) of each method in real-world applications.

Statement of completion – MUST be included in your report

Item	Completed (Yes/No/Partial)
Created sets X, Y, and Z without duplicates and showing intersections.	
AVL tree insert	
AVL tree delete	
AVL tree search	
RB tree insert	
RB tree delete	
RB tree search	
Unbalanced BST insert	
Unbalanced BST delete	
Unbalanced BST search	
Discussion comparing tree data structures	
If partial, explain what has been done	

Marking Breakdown

Description	Marks allocated
AVL tree	25%
Red-Black tree	25%
Unbalanced BST	10%
Other implementation details (e.g., creating sets X, Y, and Z without duplicates, showing intersections, etc.)	10%
Discussion comparing AVL to RBT	20%
Overall report quality	10%