Data Structures and Algorithms 2, Course Project 2024

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 31st May 2024 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 vs Skip Lists

- Create an **array** of 5,000 integers whose values start at 1 and end at 5,000.
- Implement the **Knuth shuffle** algorithm to randomise the order of the elements in the array. The implementation should yours; don't use any inbuilt array shuffling function from your programming language.
- Insert all the 5,000 integers from the array into an **AVL tree**, a **Red-Black tree**, and a **Skip List**. The implementations of these three data structures should be your own.
- Create **another array** containing 1,000 random integers in the range [0…100,000]. This array may contain duplicates.
- Insert all the elements from this **second array** into the **AVL tree**. When inserting, keep track of the following statistics:
    - The minimum, maximum, mean (and standard deviation), and median number of steps required to reach the insertion point.
    - The minimum, maximum, mean (and standard deviation), and median rotations performed. A double rotation (LR or RL) counts as one rotation.
    - The height of the tree.
    - The number of leaves in the tree.
- Insert all the elements from this **second array** into the **Red-Black tree**. When inserting, keep track of the following statistics:
    - The minimum, maximum, mean (and standard deviation), and median number of steps required to reach the insertion point.
    - The minimum, maximum, mean (and standard deviation), and median rotations performed. A double rotation (LR or RL) counts as one rotation.
    - The height of the tree.
    - The number of leaves in the tree.
- Insert all the elements from this **second array** into the **Skip List**. When inserting, keep track of the following statistics:
    - The minimum, maximum, mean (and standard deviation), and median number of steps required to reach the insertion point.
    - The minimum, maximum, mean (and standard deviation), and median promotions performed. If a single item is promoted multiple times, each time counts as a promotion.
    - The number of levels in the skip list.
- In your evaluation, use the statistics that you gathered to discuss which of the data structure you would prefer using in the real-world and, possibly, under which conditions.
- Note: you do not need to implement the delete operation on any of the data structures.

Statement of completion – MUST be included in your report

Item	Completed (Yes/No/Partial)
Created first array of integers	
Knuth shuffle	
Inserted in AVL tree	
AVL tree insertion statistics	
Inserted in Red-Black tree	
Red-Black tree insertion statistics	
Inserted in Skip List	
Skip List insertion statistics	
Discussion comparing data structures	
If partial, explain what has been done	

Marking Breakdown

Description	Marks allocated
Knuth shuffle	5%
AVL tree	20%
Red-Black tree	20%
Skip List	20%
Gathering of statistics	10%
Discussion comparing data structures	15%
Overall report quality	10%