GTU Department of Computer Engineering
CSE 222/505 - Spring 2021
Homework 7
Due date: June 12, 2021 – 23:55

PART 1:

Provide two partial implementations of NavigableSet interface:

1. Implement following methods of NavigableSet interface using skip-list
   a. insert
   b. delete
   c. descendingIterator
2. Implement following methods of NavigableSet interface using AVL tree
   a. insert
   b. iterator
   c. headSet
   d. tailSet

PART 2:

Write method that takes a BinarySearchTree and checks whether the tree is

i. an AVL tree
ii. a Red-Black tree (Suppose BinarySearchTree has a method isRed which returns a boolean value which indicates whether the root node is a red node or not.)

PART 3:

Compare insertion performance of the following data structures.
- Binary search tree implementation in the book
- Red-Black tree implementation in the book
- 2-3 tree implementation in the book
- B-tree implementation in the book
- Skip list implementation in the book

For each data structure,
- Construct an instance of each data structure by inserting a collection of randomly generated (non-repeating) numbers. Perform this operation 10 times for 10.000, 20.000, 40.000 and 80.000 random numbers (10 times for each). So, you will have 10 instances for each data structure and each different size (i.e., 200 instances in total).
- Compare the experimental run-time performance of the insertion operation for the data structures above as follows:

o Insert 100 extra random numbers into the structures you built and measure the running time
o Calculate the average running time for each data structure and problem size (i.e., number of elements in data structure).
o Draw a graph for running-time vs problem size.
o Compare the running times and their increase rate.

.

RESTRICTIONS:

- Can be only one main class in project
- Don't use any other third part library

GENERAL RULES:

- For any question firstly use course news forum in Moodle, and then the contact TA.
- You can submit assignment one day late and will be evaluated over sixty percent (%60).

TECHNICAL RULES:

- You must write a driver function that demonstrates all possible actions in your homework. For example, if you are asked to implement an array list and perform an iterative search on the list then, you must at least provide the following in the driver function:
  o Create an array list and add items to the list. Append items to head, tail, and k^{th} index of the list.
  o Perform at least two different searches by using two items in the list and print the index of the items.
  o Perform another search with an item that isn't in the array list and inform the user that the item doesn't exist in the array list.
  o Delete an existing item from the list and repeat the searches.
  o Try to delete an item that is not on the array list and throw an exception for this situation.

  The driver function should run when the code file is executed.

- Implement clean code standards in your code;
  o Classes, methods and variables names must be meaningful and related with the functionality.
  o Your functions and classes must be simple, general, reusable and focus on one topic.
  o Use standard java code name conventions.

REPORT RULES:

- Add all javadoc documentations for classes, methods, variables ...etc. All explanation must be meaningful and understandable.
- You should submit your homework code, Javadoc and report to Moodle in a "studentid_hw1.tar.gz" file.
- Use the given homework format including selected parts from the table below:

Detailed system requirements	X

The Project use case diagrams (extra points)	X
Class diagrams	X
Other diagrams	
Problem solutions approach	X
Test cases	X
Running command and results	X

GRADING :

- **No OOP design:**   **-100**
- **No interface:**   **-95**
- **No method overriding:**   **-95**
- **No error handling:**   **-50**
- **No inheritance:**   **-95**
- **No polymorphism:**   **-95**
- No javadoc documentation: -50
- No report: -90
- Disobey restrictions: -100
- **Cheating:**   **-200**
- Your solution is evaluated over 100 as your performance.

CONTACT :

- Teaching Assistant : Başak Karakaş
- bkarakas2018@gtu.edu.tr