GTU Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 7
Due date: 3 June 2020 – 23:55

This assignment consists of 4 questions.

Q1:
Build the following data structures by using the sequence of integers "20, 30, 8, 47, 39, 18, and 'last four digits of your student number in two-digit format' (i.e. 181041356 as 13 and 56)". After you build the data structure, remove all the items one by one in the same order (first in, first out).

- AVL Tree
- Red-Black Tree
- 2-3 Tree
- Skip List
- B-Tree with order 4

Show your work step by step and make your explanation. Write your solution by latex, word or handwriting (scan or take picture) and upload it as a single *StudentNumber.pdf* file.

Q2:
Modify the skip list implementation in the book so that each node in the lowest-level list keeps several elements instead of just one element as in a B-tree node.

- The maximum and the minimum number of elements in a node should be specified during the construction of the data structure. The number of elements at each node should be smaller than the maximum. The number of elements at each node should be larger than the minimum, except in the case of one node skip list.
- You should insert the new element into the corresponding node. You may need to split the node when it is necessary.
- After a deletion operation, the number of elements in a node may become smaller than the minimum. In that case, you should combine the node with its predecessor or successor.

Q3:
Compare the performance of the following data structures.

- Regular binary search tree
- Red-Black tree implementation in the book
- Red Black tree implementation in java
- B-tree implementation in the book
- Skip list implementation in the book
- Skip list implementation in java
- Skip list in question Q2

For each data structure, do the following:

- Insert a collection of randomly generated 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 of each data structure for each 4 different sizes. There should be 240 data structure in total.
- Verify that data structures are built correctly (for example, for BST, perform an inorder traversal).
- Compare the run-time performance of the insertion operation for the data structures. Insert 10 extra random numbers into the structures you built. Measure the running time and calculate the average running time for each data structure and four different problem size. Compare the running times and their increase.
- Compare the run-time performance of the deletion operation for the data structures. Perform 10 successful deletion operations from the structures you built. Measure the running time and calculate the average running time for each data structure and four different problem size. Compare the running times and their increase.

Q4:

Implement a menu-driven program for managing a software store. The system has two types of users: administrators who can update information and users who browse software. To update the information, administrators have to enter the system with a password (a single general password is sufficient). Software search is a public feature. The system maintains the information about each software which at least includes name (including version), quantity, and price. The system should be included at least the following functionalities:

- Administrator enters the system with a password, to be able to add, delete, update information.
- Search by name, quantity etc. should be available.
- The program should allow the tree to be updated when new software packages arrive at the store and when a software package is sold.
- If a package is sold out, the information about the package should be deleted.

Your implementation should use a search tree interface. So, it should be possible to use any data structure implanting search tree interface in your implementation. When it is invoked, your program should automatically create a data structure including the following software packages; Adobe Photoshop 6.0, Adobe Photoshop 6.2, Norton 4.5, Norton 5.5, Adobe Flash 3.3, Adobe Flash 4.0.

RESTRICTIONS:

- Use only specified data types
- Can be only one main class in each question
- 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 (%40).

TECHNICAL RULES:

- Use given CSE222-VM to develop and test your Homeworks (your code must be working on CSE222-VM), CSE222-VM download link will be given on Moodle.
- Implement clean code standards in your code;
    - Classes, methods and variables names must be meaningful and related with the functionality.
    - Your functions and classes must be simple, general, reusable and focus on one topic.
    - 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_hw3.tar.gz" file.
- Use the given homework format including selected parts from the table below:

Detailed system requirements	
Use case diagrams (extra points)	
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 : Ümit Murat Akkaya
- Email: umit.akkaya@gtu.edu.tr