GIT Department of Computer Engineering
CSE 222/505 - Spring 2017
Homework 6
Due date: 21.04.2017, 23:55

SCENARIO:

Q1:

Write a BinaryHeap class which extends BinaryTree class and implements Queue interface. Note that you can override any BinaryTree class structure as you need. Give detailed information (why do you need and how you modified) in your report about what you changed.

Test:

Consider the case study in Chapter 4 in your book, “Simulate Strategy for Serving Airline Passengers”. Implement the simulation with some modifications:

- Use 1 BinaryHeap instance instead of 2 PassengerQueue instances.
- Instead of using the frequentFlyerMax value, determine your passenger serving strategy by assigning a priority value to each of your passengers. The BinaryHeap will be constructed according to the priority values.

Q2:

Consider the case study in Chapter 6 in your book, “Building a Custom Huffman Tree”. Write the HuffmanTree class in the case study. Add encode method to the class structure.

Test:

Write a test program which creates a Huffman tree by reading freq.txt file. The file will include a character and the frequency value separated by a space in each line. Add some encode and decode operations to test your methods.

Q3:

Implement a level-order traverse method for the third part of the second question in Homework 5. Test your method.

RESTRICTIONS:

- Write a main class for 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 twenty percent (%20).
- Register github student pack and create private project and upload your projects into github.
- Your appeals are considered over your github project process.

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 standarts 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 standart 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_hw#.tar.gz file.
- Use the given homework format including selected parts:

Detailed system requirements	
The Project usecase diagrams (extra points)	
Class diagrams	x
Other diagrams	
Problem solutions approach	x
Test cases	x
Running command and results	

GRADING :

- No error handling : -50
- No javadoc documentation : -50
- No report : -90
- Disobey restrictions : -100

- **Cheating : -200**
- Your solution is evaluated over 100 as your performance.

CONTACT :

Nur Banu Albayrak