

**GIT Department of Computer Engineering**  
**CSE 222/505 - Spring 2015**  
**Homework 06**  
**Due date: April 12 2016 – 08:00 AM**

**PART I:**

Use HuffmanCode class which described in the text book add encode method which gets string and return its Huffman codes sequences.

Your prototype should be :

```
/**
 * Method to encode string message into Huffman encodes.
 * @param message The input message as a String
 * which is composed on the specified alphabet in the book
 * @param huffmanTree It's created huffman code for the alphabet
 * @return The encoded message as a String zero and ones.
 */
public String encode(String message, BinaryTree huffmanTree)
```

Note : Don't use table, traverse table for each character in string.

**PART II:**

Implement iterator class which iterate binarySearchTree which implemented in the book. Your iterator traverse tree ascending order.

**PART III:**

Use given priority queue interface and implement your priority queue class. Your priority queue stores any type data. To use priority property for priority queue use and implement compareto function/class which described comparable interface in java.

A priority queue can be implementing using a variety of data structures, each with different tradeoffs between memory required, runtime performance, complexity of code, etc. In this homework, you will consider four different implementations; unsorted vector, array list, linked list which implemented in java and binary search tree which implemented in the book.

Create a test class which create random data sequences (sequence size like 10, 100, 1000, 10.000, 100.000, 1.000.000) and test your four priority queue implementation running time performance. Create a table and give comparable benchmark table. (The table is important)

**OBJECTIVES:**

- Preparing object oriented design for the problem
- Creating interfaces
- Applying polymorphism
- Applying method overriding
- Applying error handling
- Applying inheritance
- Applying code documentation
- Applying clean code standards
- Creating javadoc documentation

**RESTRICTIONS:**

- Use maven standard Project template
- Use only ArrayList, LinkedList, Vector, Binary Search Tree data structure
- 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.
- Use [maven project management tool](#). And upload maven project into moodle.
- Code the Project in Java programming language. Java must be 1.8.\* or bigger version.
- Any java IDE can be used in coding process.
- Implement all interfaces class
- Add all [javadoc](#) documentations for classes, methods, variables ...etc. All explanation must be meaningful and understandable.
- Implement [clean code standarts](#) 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 standart [java code name conventions](#).
- Register [github student pack](#) and create private project and upload your projects into github.
- Your appeals are considered over your github project process.
- You can submitting assignment one day late and will be evaluated over forty percent (%40).
- Create report which include;
  - o Your name, surname, studentid
  - o Detailed system requirements
  - o The Project usecase diagrams (extra points)
  - o Class diagrams
  - o Problem solutions approach
  - o Test cases
  - o Running command and results

**GRADING :**

- No OOP design : -100
- No maven Project : -100
- No banchmar table : -100
- No interface : -95
- No method overriding : -95
- No error handling : -95
- No inheritance : -95
- No polymorphism : -95
- No javadoc documentation : -95
- No clean code standard : -95
- No report : -90
- Disobey restrictions : -98
- Your solution is evaluated over 100 as your performance. Don't forget this is performance project.

**CONTACT :**

- Teacher Assistant : Necmeddin Çarkacı
- Information and Security Lab – 122 – ncarkaci@bilmuh.gyt.edu.tr