

**GIT Department of Computer Engineering**  
**CSE 222/505 - Spring 2017**  
**Homework 7**  
**Due date: 01.05.2017, 23:55**

**SCENARIO:**

**Q1:**

Turkey have many cities and every city consists of many districts. In the first part assume that district names are unique across the country. We want to access the city name of a district within a short period of time. Consider that there may be thousands of districts. You should implement a specific (BinaryNavMap) structure that implements a navigable map with BST to do that.

**Q2:**

In the second part assume that district names may not be unique across the country. Part 1 is not capable of keeping multiple districts with the same name. In order to handle this implement a Hash Table with Chaining. In this hash table, use hash tables with open addressing instead of a LinkedList for chaining.

**Test:**

**We will send you a simple code example to test yourself. You should try to pass these tests to get full point.**

**Bonus:**

Compare performance of Q1, Q2 and ArrayList implementations using with jprofiler tools. Add your result in your report.

**RESTRICTIONS:**

- 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;
  - 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.

#### 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	
Other diagrams	x
Problem solutions approach	x
Test cases	x
Running command and results	x

#### 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