CSC 212 Project Phase II
Developing a Personal Task Manager

College of Computer and Information Sciences
King Saud University

Fall 2014

1 Introduction

In this phase of the project, two main changes are introduced. First, the time-complexity required to perform the commands OT, OG, OP, AT, AG, AP, RT, RG, and RP must be less than O(n). You are required to change the data structures, if necessary, to accommodate this new requirement. The project report should document the performance changes between Phase 1 and Phase 2, both theoretically and practically. The second change is the addition of three new commands: NT, NG and NP.

2 Command syntax

The commands are passed to the program as a string, and their results are also returned as strings. In addition to what was indicated in Phase 0 and Phase 1, Table 1 give the syntax and the meaning of the new commands.

Table 1: Query format

Query	Meaning	Example
NT ID	Count the number of key comparisons required to find the task with identifier ID. **Return string:** number of key comparisons required to find the task with Identifier ID.	NT 1278
NG ID	Count the number of key comparisons required to find the group with identifier ID. **Return string:** number of key comparisons required to find the group with Identifier ID.	NG 3008
NP ID	Count the number of key comparisons required to find the project with identifier ID. **Return string:** number of key comparisons required to find the project with Identifier ID.	NP 1019

3 Software requirements

You must implement a class named *TaskManager* that offers at least two methods [1]: *load* and *query*:

- The method *load*: takes as input the name of a file containing tasks description and load it into memory. The method returns true if the file is loaded successfully, false otherwise (**for instance, if the file is not found, the tags are incorrect, the values are illegal, a task is declared a member of a group or project but not described in the file, the file has inconsistent dependencies, etc.**). A new load operation will erase all previously loaded data. The method signature is *public boolean load(String fileName).*
- The method *query*: takes as input a string containing a command and returns the result of the command. The time-complexity of the commands OT, OG, OP, AT, AG, AP, RT, RG, and RP must be less than O(n). The method signature is *public String query(String command).*

Remark 1. *The constructor of* TaskManager *must not have any parameters. The methods* load *and* query *must not through any exception. Use the default package.*

Remark 2. *Tasks, projects and groups must be inserted in the order of their description in the file. For example, in the the file:*

```
ST
TI 2154
TS Future
ET
SG
GI 2874
GE 2145
GE 3698
GE 1478
EG
ST
TI 1478
TS Future
ET
ST
TI 3698
TS Future
ET
```

the tasks must be inserted in the order: 2154, 1478, 3698.

[1] You are free to add other methods and classes.

4 Performance Analysis

You must analyze the performance of Phase 1 and Phase 2, both both theoretically and practically:

- Analyze the performance using Big-Oh notation. Document and compare the Big-Oh for the command "OT ID" from Phase 1 implementation with Phase 2 implementation.
- Analyze the performance using timed test runs. Run Phase 1 and Phase 2 with the provided test class (TestRun) with the test files (run01.txt-run05.txt). There should be 5 test runs for Phase 1 over the files run01.txt ... run05.txt, and the same 5 test runs for Phase 2. Document the time results in a table (Table 2).
- Compare the theoretically analysis in the first part with the practical analysis in the second part for the two phases. If the results are consistent, then state whether the change in performance is worthwhile or not. If there are any contradictions between theoretically and practical analysis, then state what might be the reason.

Table 2: Test Runs

Phase/Test	run01.txt	run02.txt	run03.txt	run04.txt	run05.txt
Phase 1					
Phase 2					

5 Deliverables and rules

You must deliver:

1. A report written using the provided template. It should cover all the phases, and add the new section "Performance Analysis".
2. Source code on CD (you may be asked to submit your source code online).

The submission **deadline** is: **22/12/2014**.

You have to read and follow the following rules:

1. All data structures used in this project **must be implemented** by the students. The use of Java collections or any other library is strictly forbidden.
2. This project is to be conduct by groups of **three** students. Groups of more than three students are not accepted. Groups of two students are strongly discouraged and can only be accepted with a special permission from the course instructor.
3. All the members of a group must have the **same course instructor**.
4. All students must **submit** the list of their **group members** within one week of the announcement of this project. Once the groups are chosen, no student can change the group without a valid reason and the permission of the course instructor.

5. **Every member** of the group must participate in **all parts of the project**: designing the software, programming and writing the report. Members of the same group may receive different marks according to their participation in the project.

6. The submitted software will be evaluated automatically and/or in a demonstration to which all the group members must attend.

7. Any member of the group who fails to **attend the demonstration** without a proper excuse (consult the university and college regulations) shall receive the **mark 0** in the project.

8. In accordance with the university regulation, **cheating** in the project will be sanctioned by the **grade F**.