1 Laboratory Guide

1.1 About the laboratory

This semester, you will implement a series of algorithms and you will analyze the *correctness* and the *efficiency* of your own implementations. The implementations will have a starting point the pseudocode that is provided at the **course** and **seminary** sessions. You can write your code in C or C++ or another language that you are familiar with as long as you implement all the data structures that are required.

1.2 Laboratory Format

1.2.1 Rules

- *Attendance* is **mandatory**

- *One* absence can be recovered (the corresponding assignment <u>MUST</u> be *delivered* and *verified* the following week)

- A second absence can be recovered in the special laboratory session at the end of the semester (<u>surcharge</u>) by solving additional assignments (solving the corresponding assignment of the missed session will NOT be considered)

- If you miss more than 2 sessions, **you will not be able to attend the exam in the regular exam session**

- **IN EXCEPTIONAL CASES**, you can attend another laboratory sessions in the same week with another group *if* you announce your laboratory assistant on email and you receive his agreement; the assignment will be delivered to and verified by the laboratory assistant of the sessions you are signed up for (the assignment has to be uploaded on time and will be presented the following week at the laboratory session)

1.2.2 Grading

- The laboratory grade is equal to **30%** of the final grade

- The laboratory grade is composed of two parts: **assignments grade** - 2/3 of lab grade (that is **20%** of final grade) and **colloquy/lab exam** - 1/3 of lab grade (that is **10%** of final grade)

- Assignments:

  - Each assignment has the same weight in the calculation of the assignments grade - unweighted arithmetic mean formula is used

  - Each assignment has 4 grading thresholds for the following grades: 5, 7, 9 and 10 (the requirements for each grade can be viewed in the laboratory assignment documentation)

- Colloquy:

  - The colloquy consists of a closed book lab test in the last week of the semester (W14)
  - The colloquy must be held by each student on the computers available in the laboratory room (no personal laptops / computers)

- In order to pass the laboratory and be granted participation in the exam you need **both**:

  - Assignments grade ≥ 5
  - Colloquy grade ≥ 5

1.2.3 Assignments Delivery

- At the assessment discussion you will present: **source code, charts and runnable demo code**

- **Source code** (program.cpp) and the **charts MUST** be uploaded to Moodle, in the archive (program.zip), **before the lab session**

- *Unindented* code will **not be evaluated**

- If you cannot explain the used algorithms, the assignment will **not be evaluated**

- Each source file must contain at the beginning a comment of the following format:

```
/**
* @author John Smith
* @group 30xyz
*
* Assignment requirements, ex: Compare the sorting methods X and Y
*
* Personal interpretation of the (time and space) complexity of the average,
best and worst testing Interpretarea
* cases. For example: "Method X has a time complexity of Y in case Z
because ..."
*/
```

1.2.4 Algorithm complexity evaluation

- For the average case, you need to repeat the measurements *at least 5 times*

- Measure the number of operations made by the algorithm (the assignments and the comparisons of the input data or the auxiliary variables that contain input data)

- Vary the dimensions of the input data consistent with the requirements of each assignment

- Apply the same input data on each algorithm when making comparative evaluations (for the average case)

- Generate evaluation charts (either in **Excel** or by using **Profiler**)

- Analyze the charts and add your personal observations at the beginning of the source code using the above mentioned format

1.2.5   Delivery deadline

Assignments can be delivered:

- During the laboratory sessions that they are presented in. At the end of the laboratory you MUST upload to Moodle a draft version of the current assignment (containing what you managed to implement during the session). A lack of a submission at the end of the session (or a lack of relevant code) will receive of penalty of up to 2 points of the grade of that assignment

- **Extension_1** (E1): at the beginning of the next laboratory session

- **Extension_2** (E2): specific assignments can be delivered at the beginning of the second laboratory after the assignment was presented (with a penalty of -2).

- Starting with the third laboratory after the assignment was presented, the assignment cannot be delivered anymore (will be graded as 0)

You can find a planning of the assignment and their extensions on Moodle (file name: Planificare Saptamanala (bachelor))

Each assignment MUST be uploaded to Moodle before the beginning of the laboratory session during which it will be delivered.

1.2.6   Attempted Fraud

For the first uncovered attempt at fraud (copying someone else's code or using code generated by AI tools) you will receive a penalty of 10 points of the total accumulated points (until that moment). For a subsequent attempt, you will have to **retake the course next year**.

1.3   Laboratory session transfer

If you wish to participate at laboratory sessions with another assistant you must follow these rules:

- Student S1 from group G1 can transfer to group G2 if and only if a student S2 from group G2 can be found that is willing to participate with G1 at the assigned laboratory session hours of G1.

To formalize the transfer you must send an email in which you mention with whom are you making the transfer/exchange:

- An email from S1 to both laboratory assistants

- An email from S2 to both laboratory assistants

The laboratory session transfer deadline is the **end of the second week** of the semester.

1.4 Suggested literature

- Cormen, T. H. et al (2009). Introduction to algorithms. MIT press

- J. Kleinberg, E. Tardos (2005). Algorithm Design. Addison Wesley

- C/C++ Tutorials

    - http://www.cprogramming.com/begin.html
    - http://www.learn-c.org
    - Accelerated C++: Practical Programming by Example

- Coding Styles

    - http://users.ece.cmu.edu/~eno/coding/CCodingStandard.html
    - http://www.cs.swarthmore.edu/~newhall/unixhelp/c_codestyle.html
    - http://google-styleguide.googlecode.com/svn/trunk/cppguide.xml

1.5 Profiler

The library that will be utilized for the generation of plots, each student must go through the given example and tutorial.

The most recent version can be found here:
https://github.com/cypryoprisa/utcn-fa-profiler

Profiler Tutorial:
- Part 1
- Part 2
- Part 3