UNIVERSIDAD DE COSTA RICA

Escuela Ciencias de la Computación e Informática

CI-1221 Data Structures and Algorithms Analysis

I semester 2019

Homework 1

Due dates:
Part I: Friday April 12th
Part II: Friday May 7th

1 Goal

The goal of the homework is to implement the sorting and median-search algorithms studied in the first quarter of the course and verify that theoretical differences in efficiency correspond to reality.

2 Algorithms

The algorithms to be implemented are the following: (i) selection sort [7 pts.], (ii) insertion sort [7 pts.], and (iii) merge sort [7 pts.] for **part I**, and (iv) heapsort [7 pts.], (v) quicksort [7 pts.], and (vi) radix sort with base $2^{\lfloor \lg n \rfloor}$digits [10 pts.], and (vii) median search [7 pts.] for **part II**.

3 Comparison

The steps to compare the algorithms are the following:

1. The algorithms must be compared using arrays of randomly chosen integers of sizes 100.000, 200.000, 300.000, and 400.000. Execute every algorithm at least three times on each of the arrays.[1] Report these times and their mean in a table [7 pts.].

[1] If the largest execution time is at least 1.5 times the smallest, the operating system was probably doing something else while running your task. If this happens ignore that execution and run the algorithm again. Repeat as necessary until the largest execution time is no more than 1.5 times the smallest.

2. Plot mean execution times versus array size for each of the algorithms and analyze the resulting curve to identify if the shape is as expected (i.e., if the curves of the $\Theta(n^2)$ algorithms have a parabolic shape and the curves of the $\Theta(n)$ and $\Theta(n \log n)$ algorithms are almost straight).[2] (Do not forget to indicate the units used in the time axis, i.e., s, ms, etc.) [7 pts.].

3. Plot the curves again all together in a new set of axis. Show the time in a logarithmic scale to counteract the large execution time differences between the quadratic and (almost) linear algorithms. Analyze if the relation between the curves is as expected [7 pts.].

This process must be explained and analyzed in detail in the homework report (see Section 4.1). For example, it must be reported if searching for the median using the specific algorithm is faster than sorting the whole array (with the fastest algorithm) and then picking the number in the middle.

4 Deliverables

The homework has two parts. The first one is a preliminary report that analyzes the selection sort, insertion sort, and merge sort algorithms. The second part analyzes the heapsort, quicksort, radix sort (with base $2^{\lfloor \lg n \rfloor}$ digits), median search, and the former three sorting algorithms. Also, the code of the algorithms must be included in the file *Ordenador.h* published with this description. An optional *style checklist* can also be submitted to claim extra credit for style (see Section 4.1.4). The following sections explain each of the deliverables.

4.1 Reports

The following sections describe the expected content, languages, format and style of the reports.

4.1.1 Contents

The parts of the reports must be those of a scientific article. They must contain (at least) the following sections: title, abstract, introduction, methodology, results, conclusions, and bibliography. However, the titles of the sections can change if necessary. Next, we describe the content of each section.

Title The title must be representative of content content of the paper. Avoid using interrogative titles like *What is the best sorting algorithm?* [1 pt.]

Abstract An abstract summarizes the content of the paper in a few lines (a paragraph, to be exact). It summarizes the research questions, methodology used to answer it, results obtained, and main conclusions you arrived to (hopefully, the answer

[2] Recall that $n \log n < n^{1+\epsilon}$ for $\epsilon > 0$ y n sufficiently large n.

to the questions). Each section of the report is summarized here in one or two sentences to form a paragraph with four to eight sentences. The abstract is the first thing in the paper (after the title and authors' info), but is the last thing to be written (for obvious reasons). [4 pts.]

Introduction The Introduction presents the research questions and a brief description of what will be done (the methodology) to answer them.[3] The predominant form of communication in the Introduction is text. Cites to bibliographical references are usually placed here.[4] [5 pts.]

Methodology The Methodology presents the details of the experiments that permit to answer the formulated questions. In this section, a detailed description of the procedure should be the predominant component. [10 pts.]

Results This section shows the results obtained from the experiments. Tables and figures are usually abundant here. [21 pts. See details in Section 3]

Conclusions This section states the conclusions that can be inferred from the results.[5] Text is the dominant form here. [5 pts.]

Bibliography This section shows the information of the bibliographical sources cited in the text.[6] [2 pts.]

4.1.2 Language

Knowing the importance of the English language in students' academic and future professional performance, the homework can be written in English. However, the names of the methods in the code must be written in Spanish (see Section 4.2).

4.1.3 Format

The reports must follow the format of the Institute of Electrical and Electronics Engineers for its TRANSACTIONS journals (except the TRANSACTIONS ON MAGNETICS and TRANSACTIONS ON DIELECTRICS, which use other formats). The goal is that the student, besides learning to make technical and scientific reports, have the experience of limiting to a preestablished format (a common practice not only in academia but also in industry). Specifications of the format and LaTeX and MS-Word templates can be found in `https://www.ieee.org/publications_standards/publications/authors/author_templates.html`. LyX also provides a template with this format. (Go to File ▷ New from template and select `IEEEtran-Journal.lyx`. But be careful… some templates have similar names but different formats.)

[3] For this homework, the question should be whether the theory corresponds to practice (i.e., whether the asymptotically faster algorithms in theory, are so in practice).

[4] For this homework, a reference to the textbook is probably sufficient.

[5] Here you are expected to tell what are the best algorithms.

[6] For this homework, probably only the textbook.

4.1.4 Style

The instructor of this course promotes the use of good style practices in writing reports by giving *extra credit* for following certain typographical rules, among them: proper use of the cursive in mathematical environments (i.e., equations), proper formatting of decimals, use of *floats* for figures and tables and referencing them properly, homogeneous use of fonts, and use of word division at the end of the line to avoid excessive inter-word space (MS-Word tends to be in disadvantage on this with respect to LaTeX and LyX).

The student interested in claiming credit for style must fill and attach the to the homework the form *Style checklist.* This form is a spreadsheet that computes the credit for style automatically, based on the rules marked by the student. The form is available in Numbers, Excel, and LibreOffice formats.

The following are examples of rules:

1. Single-letter variables, constants, or functions are written italicized or in cursive (x, n, i), everything else in Roman (i.e., with no tilt: log, max, cos, etc.). Parentheses never lean.

2. Pages are numbered (except maybe the first).

To claim that a rule has been *followed*, it has to be true in at least 90% of the cases. For example, to claim that Rule 2 was followed, at least 90% of the pages must be numbered (except maybe the first). If you claim to have followed a rule but did not in at least a 90% of the cases, not only the points will be taken off, but also you will be penalized with the same number of points (i.e., the *value* of the item will be switched to negative).

4.2 Code

The code must be written in the C++ programming language, using the provided templates. The methods' headers must not be modified since the teaching assistants will use a script to run them, and any change in the methods interface will make the script to fail. (However, you can add customized private methods and call them in the body of the required methods.) To avoid compilation errors, the language must be used in its standard form. In particular, the code must compile if `g++` and the standard C++ libraries are used. *If the code does not compile you will receive a grade of zero. If a particular algorithm it does not correctly sort the arrays or find the median, you will receive a grade of zero for that algorithm.* To determine if a sorting algorithm is correct, we will run the following test:

```
for(int i=1; i<n; i++)
    if( A[i] < A[i-1] )
        cout << "Failed!";
```

To determine if the median-search algorithm is correct, we will sort the array and compare the the three elements in the middle with the one found by your algorithm. If the latter is below the minimum of those three or above the maximum, it will be considered a fail.

5 Submission

This homework must be submitted before its due date through MEDIACION VIRTUAL. The student is responsible for checking the integrity of the submitted homework, specially if it is in a zipped file.[7] If you submitted multiple copies, only the last one will be considered. If the homework is submitted after the due date, the following police (specified in the syllabus) will be applied: *the grade received by a student who submitted a homework late will not be higher than the lower grade received by a student who submitted the homework on time.*

If you are having problems to submit the homework and the due date is approaching, send it to the assistant (`denisse.alfaro@ucr.ac.cr` with copy to the instructor (`arturo.camacho@ucr.ac.cr`). Please write the following in the subject: *Tarea de Algoritmos.* In case we receive multiple submissions by email, we will consider only the first one.[8] The *late submission* policy mentioned in the previous paragraph applies to email submissions as well.

[7] Download the submitted file and unzip it to verify the integrity of its contents.

[8] The purpose is to discourage multiple email submissions.