National University of San Agustin
System Engineer School

ADA - Lab 04 - Sorting

Christian E. Portugal-Zambrano

October 10, 2018

1 Introduction

Given a :
Input: A sequence of n numbers $\langle a_1, a_2, a_3, ..., a_n \rangle$.
Output: A permutation (reordering) $\langle a'_1, a'_2, a'_3, ..., a'_n \rangle$ of the input sequence such that $a'_1 \leq a'_2 \leq a'_3 \leq ... \leq a'_n$ The input sequence is usually an n-element array, although it may be represented in some other fashion, such as a linked list [1].
We must consider these notions:

- In practice, the numbers to be sorted are rarely isolated values.
- Each is usually part of a collection of data called a record.
- Each record contains a key, which is the value to be sorted. The remainder of the record consists of satellite data, which are usually carried around with the key.
- In practice, when a sorting algorithm permutes the keys, it must permute the satellite data as well. If each record includes a large amount of satellite data, we often permute an array of pointers to the records rather than the records themselves in order to minimize data movement.

2 Pre-requisites

You must have read Chapter 2 and 6 from [1], be sure to understand basic algorithm for sorting such as bubble and selection sort. The scripts for make algorithm time analysis

from week 02 are already available, please make sure to elaborate plots with R (suggested) but you can use Pyplot for python. For this practice you will need a library to make graphics, you can use the native java libraries or Qt Graphics for C++, Opengl or VTK can be selected too.

3 Programming environment

You can use all the tools described on practices before, but here you will need some graphics tools, I recommend to use Netbeans for Java or Qt for C++, both tool brings up with a graphics library enough for our practices objectives.

4 Algorithms presentation

So, as we are working with LaTeX algorithms presentation will be easy, you just need to add this at the header of your document:

```
\usepackage{algorithm}
\usepackage{algorithmicx}
\usepackage[noend]{algpseudocode}
```

4.1 Basic Sorting

We shall address numerous algorithms for sorting arrays of objects having a natural order. To compare and contrast the algorithms, we shall examine a number of their properties, including the number of compares and exchanges that they use for various types of inputs and the amount of extra memory that they use. These properties lead to the development of hypotheses about performance properties, many of which have been validated on countless computers over the past several decades. Here you must present an analysis of running time for bubble, selection, insertion and merge sort with 1000 and 10000 elements, the elements will have two configurations: Ordered and random data, for each configuration you will have time achieved, you must perform a total of 100 samples for each configuration and plot the result using your preferred tool.

4.2 Warm-up: approach

In this section we are introducing some tools to perform graphics. Please pay attention to the explanation given for the instructor.

4.2.1 To Do:

We are trying to understand the behavior of the sorting algorithms, so it's necessary to make an analysis of these algorithms with some kind of ordered and random data, the idea behind of this is try to reproduce the presentation from here Sorting Visualization Implement these algorithm using the IDE and language of your preference, make a GUI

using Java or Qt, is up to you the colors or disposition of the widgets, but at least it would have a button to run the simulation for every configuration, you must implement the basic algorithms there.

5 Deep inside

This section is just for anyone who wants to make a deep inside into the theory and like challenges[1], you will have to prepare a presentation of just 5 minutes to explain and run the algorithm, then you will have to defend yourself another five minutes of questions. I want that all students benefit from your presentation, remember that we are here to learn. If you want to do this please email to cportugalz@unsa.edu.pe

6 Deadline

For this practice one score will be taken at class time at nearly October 24. Remember that plagiarism must be avoided and if it is detected the grade will be zero and repetition informed to superior authorities. The work can be realized with up to 3 people and a demo must be presented. All question and doubts must be done to the same email.

References

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[1] This must not be included into the report