National University of San Agustin
System Engineer School

ADA - Lab 01 - Introduction to Algorithms

Christian E. Portugal-Zambrano

August 27, 2018

1 Introduction

Welcome to Analysis and Design of Algorithms, first you must keep in mind that algorithms are not easy, you need to try and try to understand some of them, so keep trying and you will do well. In this Laboratory you will learn about some concepts and tools that will help you through the course.
Before get started we must know these important phrase:

"An algorithm must be seen to be believed" - Donald Knuth

If you don't know who was Donald Knuth, here is your first question ¿Could you tell me about who were Donald Knuth, Linus Torvalds, Dennis Ritchie, Niklaus Wirth, Prim, Kruskal, R. Sedgewick?

2 Pre-requisites

You need some experience programming, but you will learn more on class, the selection of a programming languages is a must, as your instructor recommend C, C++ or python language, this way you would avoid some OOP concepts unnecessary for design and programming algorithms. Data structures, loops, functions, recursion and some math libraries will be required to this course.

3 Programming environment

Don't worry much about this, as your instructor I will not force you to choose a specific one, contrary to this I encourage you to select from these or another one you are already selected:

- Eclipse - https://www.eclipse.org
- Netbeans - https://netbeans.org
- Anaconda - https://www.anaconda.com
- Visual Studio (Code) - https://visualstudio.microsoft.com/es/
- Jetbrain IDE's - https://www.jetbrains.com
- Atom - https://ide.atom.io
- Sublime - https://www.sublimetext.com

All IDE's I've mentioned above are multi-platform and GUI based, but if you are a console user probably you will feel comfortable with VIM or EMACS.

4 Algorithms presentation

So, this is becoming interesting, for all your work at least you will get in touch with an algorithm, but ¿What is an algorithm? ... yes, you will find the answer at google.
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}
```

Let's get start with the nth root problem!

4.1 Nth root problem

Given a root, a decimal number and a error (epsilon), we need to build a function that give us an approximated nth root considering the error rate. The complexity of the algorithm must be logarithmic.

4.2 Some related algorithms

This kind of algorithms or numeric methods are used to find a value x so that $f(x) = 0$ where f is a continuous function, there are algorithms like Newton's method, Halley, Ruffini's rule, and so forth. Here you will use the bisection algorithm to find the roots of a equation given.

4.2.1 Bisection algorithm

It is an iterativo method used to find the roots of non linear equations based on find a solution of $f(x) = 0$ throughout a sequent of decreasing intervals as shown in [1], here the author explains that for a continuous function defined between an interval $[a, b]$ and $f(a)$,$f(b)$ functions inside the interval, we can find at least one root inside the interval. So we calculate $c = \frac{a+b}{2}$ and $f(c)$, if $f(c)$ is not a root then the new interval is given by the function with opposite signs, this process is repeated until the error rate is achieved or the iterations given finished, these iteration are needed to avoid a infinite loop. The bisection algorithm of [2] is presented in the Algorithm 1

Algorithm 1 Bisection algorithm to find the nth root

Require: $a < b$, ($f(a) < 0$ and $f(b) > 0$) or ($f(a) < 0$ and $f(b)) > 0$
Ensure: nth root of x

```
1:  procedure NthRootbyBisection(nth, x, error, iteration)
2:      N ← 1
3:      while N ≤ iteration do
4:          c ← (a+b)/2
5:          if f(c) = 0 or (b−a)/2 < error then
6:              Return(c)
            N ← N + 1
7:          if Sign(f(c)) == Sign(f(a)) then
8:              a ← c
9:          else
10:             b ← c
```

You must observe that $f(x) = \sqrt[n]{x}$, then you can consider a variable replacement so that $y = \sqrt[n]{x}$, resolving you will have the objective function $0 = y^n - x$, you will have to find the roots of that equation. To avoid problems with imaginary number a and b must be between the range of 0 and x.

5 Warm up

You must analyze the input of the algorithm, consider to check all the initial conditions before start to program it. Here you have to run the algorithm by hand, try to understand it, once you do that try to explain it. I hope you include this explanation into your report, remember to be concise, simple and original.

6 To do

After you understand the algorithm, you must implement it using the programming language and tools of your preference, you must include your implementation into the report and try to ask this questions:

- ¿How much iteration you need? ¿Is it a fixed number? Explain.
- ¿What programming language you choosed and why?
- ¿What is the average cost of the algorithm implemented? ¿What is the unit to measure cost?
- ¿Are there a relation between the theory cost and the implementation cost?
- Find another algorithm to resolve the nth root problem, explain it and make a comparison between both.

7 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

8 Deadline

For this course we have three groups, according to this the deadline for each group is six days after the lab group scheduled. Remember that plagiarism must be avoided and if it is detected the grade will be zero and reincidence informed to superior authorities. An pdf unnamed and named report must be sent to cportugalz@unsa.edu.pe before deadline, after deadline you will be qualified under the minimum grade obtained with a maximum of thirteen and penalized with 1 point per day late. All question and doubts must be done to the same email.

References

[1] S. D. Conte and C. W. D. Boor, *Elementary numerical analysis: an algorithmic approach*. McGraw-Hill Higher Education, 1980.

[2] R. L. Burden and J. D. Faires, *Numerical Analysis*. PWS Publishers, 3rd ed., 1985.

[1] This must not be included into the report