Carlo Ángel Luján García

A01639847

Análisis y diseño de algoritmos avanzados (Gpo 601)

Programming Lab 01

09/05/2022

Tecnológico de Monterrey

Programming Lab 01

Objectives:

- Empirically establish the intuition of the relation between the size of a problem and the change in the computational cost.
- Get familiar with the Linux/terminal compilation.

Summary:

In the following document we'll talk about the programming Lab 01 results, which consist of the established intuition of the relation between the size of a problem and the change of computational cost in an arithmetic sum problem.

In the proposed problem we want to find the arithmetic sum of the elements inside a numeric array/vector of multiple sizes.

Example:

Given an array of size 4: [1,2,3,4] we sum each of the numbers inside of it: 1 + 2 + 3 + 4 and we should get the number 10.

We want to find if a relation exists in computational cost compared to the size of the problem, meaning, in this case, the cost of more iterations to get to the arithmetic sum solution.

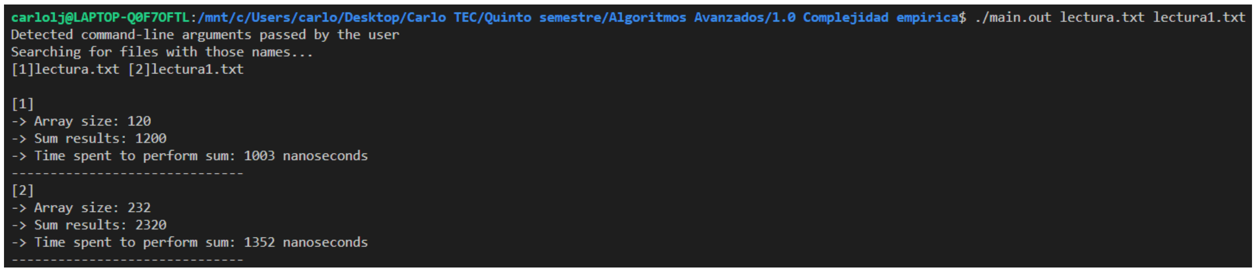
The code:

Following the instructions in the given lab document the first thing that we want to get rid of is the capability to receive command-line arguments passed by the user, including the name of the program, and single or multiple filenames contained in the argument vector.

If we pass a command-line argument value, the program should detect it and search for a file (in the application containing folder) with the same name of the argument passed using the command-line.

In case we have more than one “argc” (ARGument Count) this will mean that our argv (ARGument Vector) contains more data. We should get all the “filenames” inside the argument vector and search for those files, and if the files exist, we must get all of their corresponding number vectors, effectuate the arithmetic sum process, and get the elapsed time to perform the sum operation.

Example:



```
carlolj@LAPTOP-Q0F7OFTL:/mnt/c/Users/carlo/Desktop/Carlo TEC/Quinto semestre/Algoritmos Avanzados/1.0 Complejidad empirica$ ./main.out lectura.txt lectura1.txt
Detected command-line arguments passed by the user
Searching for files with those names...
[1]lectura.txt [2]lectura1.txt

[1]
-> Array size: 120
-> Sum results: 1200
-> Time spent to perform sum: 1003 nanoseconds
------------------------------
[2]
-> Array size: 232
-> Sum results: 2320
-> Time spent to perform sum: 1352 nanoseconds
------------------------------
```

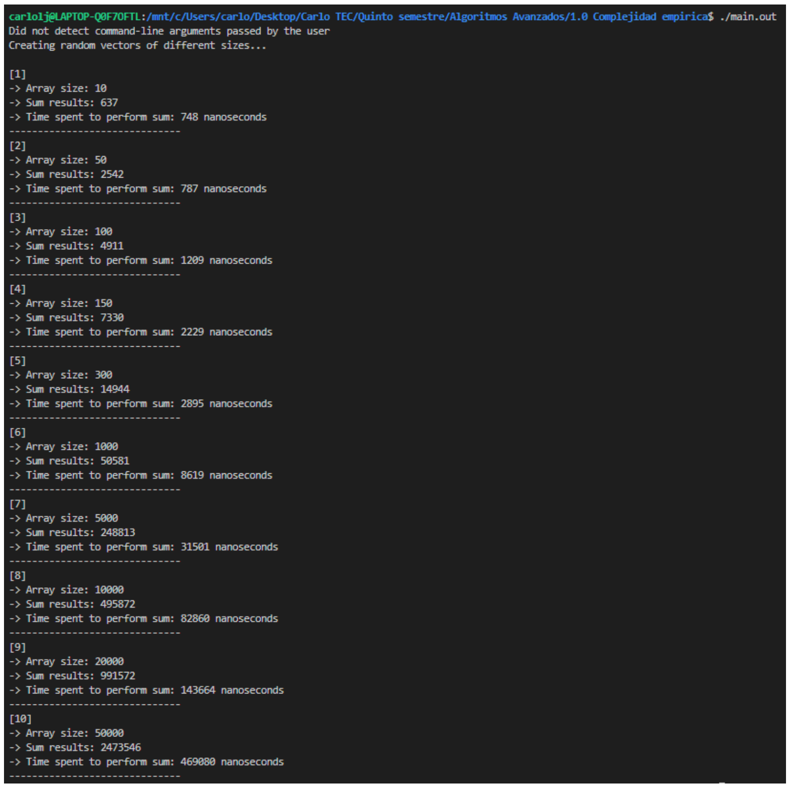
The program should also allow a case where the user didn’t enter any extra command-line arguments. This case will trigger a default case that will generate random integer vectors of higher and higher sizes to demonstrate that the execution time of the current problem will increase over the size of the vector of numbers to sum, meaning more iterations needed to complete the sum.

This default case will use the following 10 test cases, where the numbers in the array represent the number of random integers to generate for the sum. We expect to have higher computational cost in the higher sized number arrays.

```
int m[10] = {10, 50, 100, 150, 300, 1000, 5000, 10000, 20000, 50000};
```

Example:

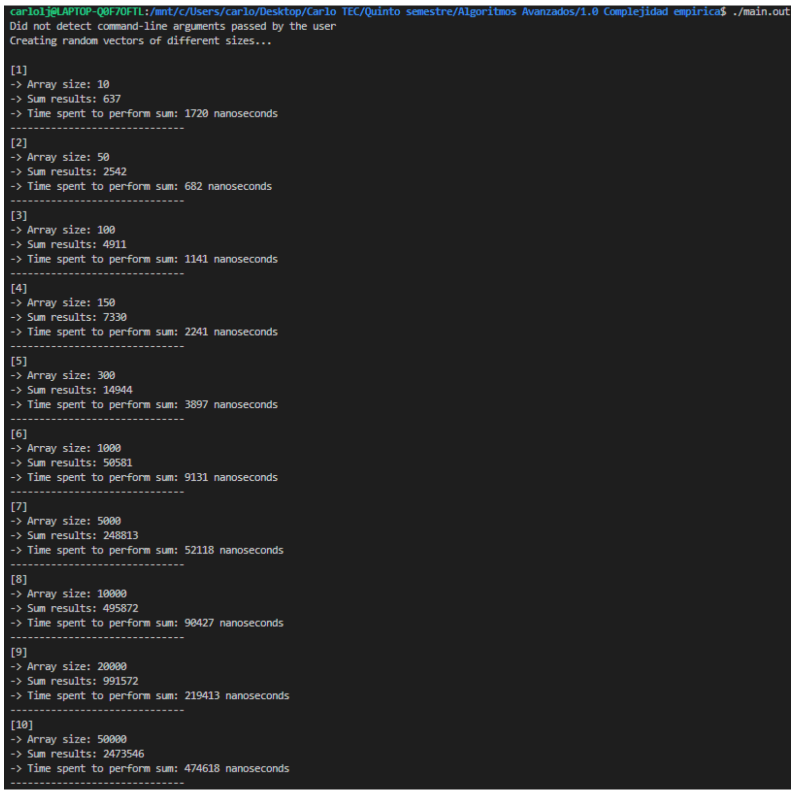
Run [1]



```
carlolj@LAPTOP-Q0F7OFTL:/mnt/c/Users/carlo/Desktop/Carlo TEC/Quinto semestre/Algoritmos Avanzados/1.0 Complejidad empirica$ ./main.out
Did not detect command-line arguments passed by the user
Creating random vectors of different sizes...

[1]
-> Array size: 10
-> Sum results: 637
-> Time spent to perform sum: 748 nanoseconds
------------------------------
[2]
-> Array size: 50
-> Sum results: 2542
-> Time spent to perform sum: 787 nanoseconds
------------------------------
[3]
-> Array size: 100
-> Sum results: 4911
-> Time spent to perform sum: 1209 nanoseconds
------------------------------
[4]
-> Array size: 150
-> Sum results: 7330
-> Time spent to perform sum: 2229 nanoseconds
------------------------------
[5]
-> Array size: 300
-> Sum results: 14944
-> Time spent to perform sum: 2895 nanoseconds
------------------------------
[6]
-> Array size: 1000
-> Sum results: 50581
-> Time spent to perform sum: 8619 nanoseconds
------------------------------
[7]
-> Array size: 5000
-> Sum results: 248813
-> Time spent to perform sum: 31501 nanoseconds
------------------------------
[8]
-> Array size: 10000
-> Sum results: 495872
-> Time spent to perform sum: 82860 nanoseconds
------------------------------
[9]
-> Array size: 20000
-> Sum results: 991572
-> Time spent to perform sum: 143664 nanoseconds
------------------------------
[10]
-> Array size: 50000
-> Sum results: 2473546
-> Time spent to perform sum: 469080 nanoseconds
------------------------------
```

Run [2]



```
carlolj@LAPTOP-Q0F7OFTL:/mnt/c/Users/carlo/Desktop/Carlo TEC/Quinto semestre/Algoritmos Avanzados/1.0 Complejidad empirica$ ./main.out
Did not detect command-line arguments passed by the user
Creating random vectors of different sizes...

[1]
-> Array size: 10
-> Sum results: 637
-> Time spent to perform sum: 1720 nanoseconds
------------------------------
[2]
-> Array size: 50
-> Sum results: 2542
-> Time spent to perform sum: 682 nanoseconds
------------------------------
[3]
-> Array size: 100
-> Sum results: 4911
-> Time spent to perform sum: 1141 nanoseconds
------------------------------
[4]
-> Array size: 150
-> Sum results: 7330
-> Time spent to perform sum: 2241 nanoseconds
------------------------------
[5]
-> Array size: 300
-> Sum results: 14944
-> Time spent to perform sum: 3897 nanoseconds
------------------------------
[6]
-> Array size: 1000
-> Sum results: 50581
-> Time spent to perform sum: 9131 nanoseconds
------------------------------
[7]
-> Array size: 5000
-> Sum results: 248813
-> Time spent to perform sum: 52118 nanoseconds
------------------------------
[8]
-> Array size: 10000
-> Sum results: 495872
-> Time spent to perform sum: 90427 nanoseconds
------------------------------
[9]
-> Array size: 20000
-> Sum results: 991572
-> Time spent to perform sum: 219413 nanoseconds
------------------------------
[10]
-> Array size: 50000
-> Sum results: 2473546
-> Time spent to perform sum: 474618 nanoseconds
------------------------------
```

To run this program and get the same results as shown, it’s recommended to use the following lines in your Linux terminal:

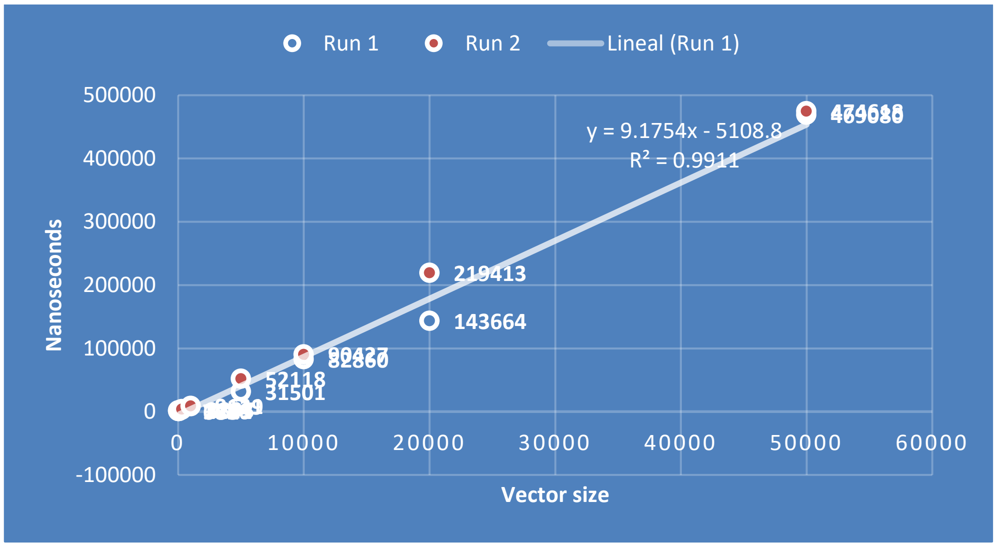
```
➔ g++ -std=c++17 A01639847_Actividad_1_0_Complejidad_empirica.cpp -o main.out
```

and then:

```
➔ ./main.out
  or
➔ ./main.out <existing_filename1> <existing_filename2> … <existing_filenameX>
```

The results:

Based on our last picture results we can plot our data and find the relation between the size of a problem and the change in the computational cost (execution time in nanoseconds).



As we can see the tendency is lineal, meaning that follows the function $f(x) = mx + b$ where m is the slope and b is a constant. A lineal tendency function tells us that given a variable “x” (in this case our vector size) the “y” value will increase by a factor of m +/- the constant b. Since our “y” variable means nanoseconds on execution time we can confirm that when the vector size increases the program will need more time to finish the execution.

Device used:

Model: Asus TUF Dash F15
SO: Windows 11 Home with WSL
Processor: 11th Gen Intel Core i7-11370H 3.30 GHz
RAM: 16.0 GB (15.7 GB utilizable)