

## Laboratory practice No. 2: Algorithms Complexity.

**Samuel David Ben Jacob Arango Henao**  
Universidad Eafit  
Medellín, Colombia  
Sdarangoh@eafit.edu.co

### 3) Practice for final project defense presentation

#### 3.1. Merge Sort.

N(millones)	Times (ms)
10	1115
20	2124
30	3077
40	4483
50	7076
60	7898
70	7999
80	8552
90	10110
100	45210
110	13178
120	14335
130	34585
140	19672
150	20527
160	30360
170	31547
180	23289
190	23232
200	24043

#### Insertion Sort

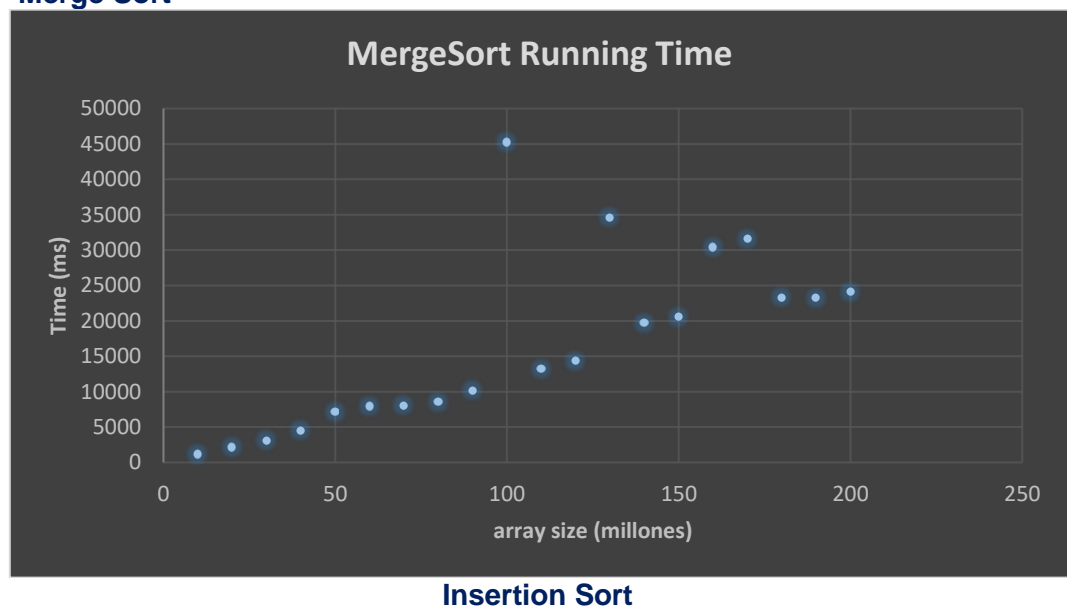
**PhD. Mauricio Toro Bermúdez**  
Professor | School of Engineering | Informatics and Systems  
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627  
Phone: (+57) (4) 261 95 00 Ext. 9473

## ESTRUCTURA DE DATOS 1

### Código ST0245

N(mil)	Times (ms)
1	9
2	3
3	48
4	56
5	47
6	13
7	15
8	18
9	30
10	35
11	43
12	43
13	47
14	59
15	55
16	62
17	71
18	107
19	93
20	103

### 3.2. Merge Sort



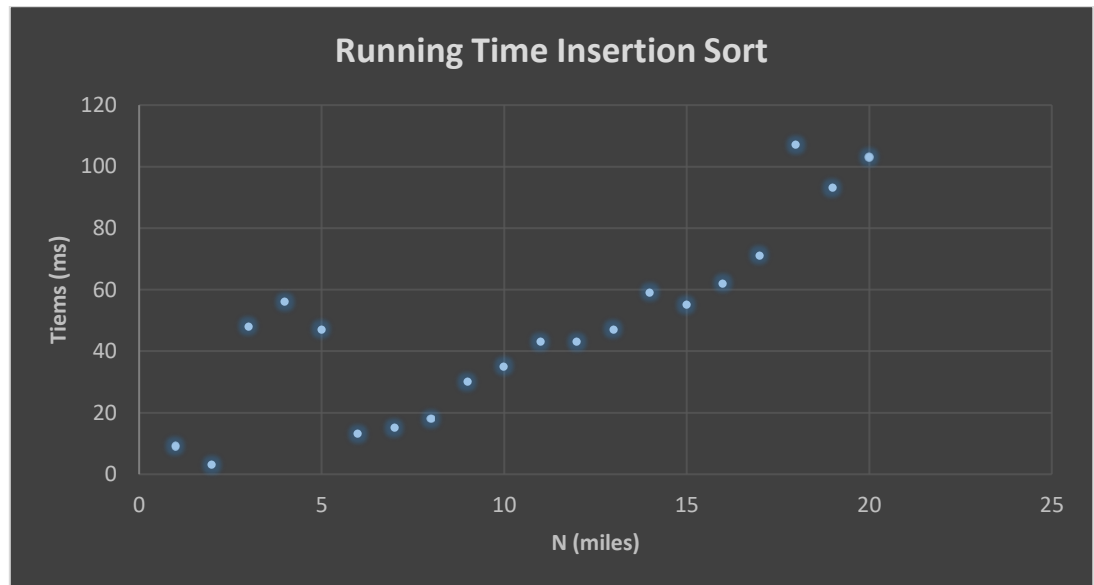
**PhD. Mauricio Toro Bermúdez**

Professor | School of Engineering | Informatics and Systems

Email: mtorobe@eafit.edu.co | Office: Building 19 – 627

Phone: (+57) (4) 261 95 00 Ext. 9473

**ESTRUCTURA DE DATOS 1**  
**Código ST0245**



**3.3.** We can see that the running time of Insertion sort increases by a factor of the square of the number of elements in the array, in that order of ideas, for a million of elements in the array, it's is going to take a time to the order of  $10^{12}$ ms which is almost 32 years, this tell us that the algorithm Insertion Sort is not efficient for videogames which use arrays with a big number of elements that must be sorted fastest as possible.

**3.4.** The Merge Sort algorithm is the type divide and conquer, that means in each step the problem is less and less complex. Usually the algorithms of this type are solved by the Master Theorem for Divide and Conquer, and the Merge Sort algorithm corresponds to the 3<sup>rd</sup> case numeral a, in that case the complexity is expressed in with a logarithm.

**“Data Structures and Algorithms Made Easy: Data Structures and Algorithms Puzzles” By Narasimha Karumanchi.**

**3.5.** When the elements of the array are already ordered the Insertion Sort algorithm will have a linear complexity, because it is not going to found elements smaller on the right side of one element. On the other hand, we have that Merge Sort will keep dividing the array, which will keep the linear logarithmic form of the complexity.

**3.6.** For each number on the array, the algorithm searches another equal and counts each number between them, saving them in a variable, and making a comparison with the previous maximum value which is initialized in cero, in case that the present value is bigger than the previous, the maximum changes and the algorithm keep searching another value bigger than that until finish in the last element of the array, and returning the final value which gets established in the max variable.

**PhD. Mauricio Toro Bermúdez**

Professor | School of Engineering | Informatics and Systems

Email: mtorobe@eafit.edu.co | Office: Building 19 – 627

Phone: (+57) (4) 261 95 00 Ext. 9473

**ESTRUCTURA DE DATOS 1**  
**Código ST0245**

**4) Practice for midterms**

**4.1** C

**4.2** D

**4.3** B

**4.4** B

**4.5** D, YES.

**4.6** 10.000s

**4.7** 1, 3, 4

**4.8** B

**4.9** A

**4.10** C

**4.11** C

**4.12** B

**4.13** C

**4.14** C

**PhD. Mauricio Toro Bermúdez**

Professor | School of Engineering | Informatics and Systems

Email: mtorobe@eafit.edu.co | Office: Building 19 – 627

Phone: (+57) (4) 261 95 00 Ext. 9473

