

Laboratory practice No. 2:

Simon Lopera
Universidad Eafit
Medellín, Colombia
sloperab1@eafit.edu.co

Juan Jose Escobar
Universidad Eafit
Medellín, Colombia
jjescobarb@eafit.edu.co

3) Practice for final project defense presentation

3.1

n	t	n	t
10000000	515	50000	803
20000000	1048	51000	810
30000000	1473	52000	1007
40000000	1998	53000	726
50000000	2523	54000	1194
60000000	2992	55000	1524
70000000	3476	56000	2230
80000000	2995	57000	1863
90000000	4506	58000	2396
100000000	5211	59000	3303
110000000	5718	60000	1467
120000000	6145	61000	1021
130000000	6553	62000	1048
140000000	7059	63000	1034
150000000	7587	64000	1084
160000000	8253	65000	1343
170000000	8832	66000	1534
180000000	9267	67000	1357
190000000	10066	68000	2051
200000000	10321	69000	3683

Merge and Insertion Sort Tables

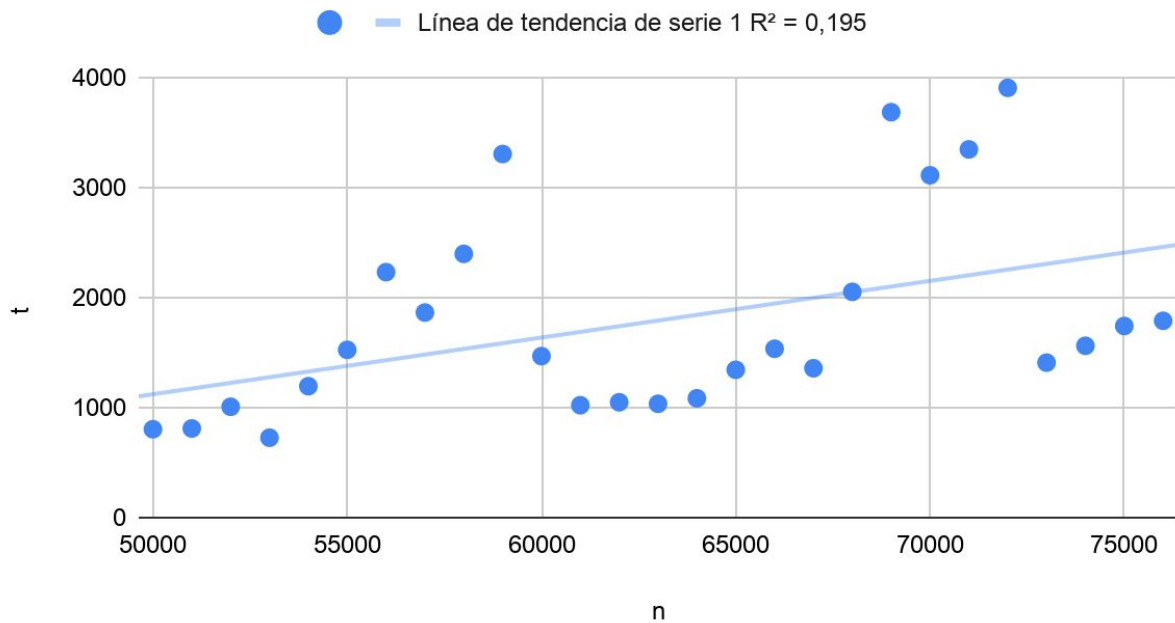
3.2

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

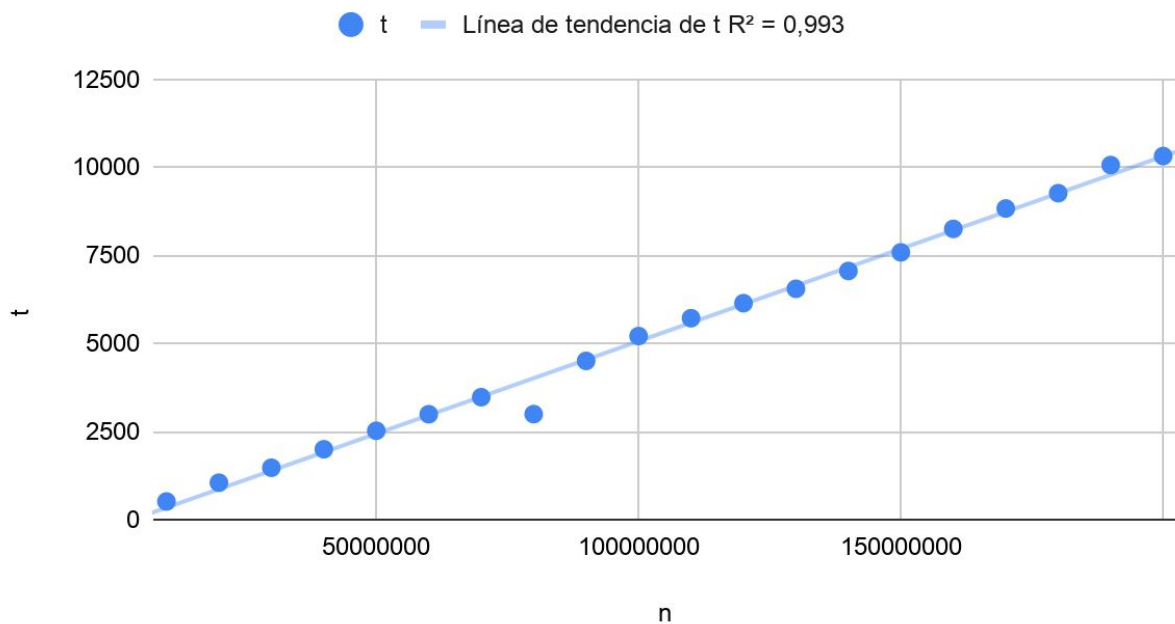
ESTRUCTURA DE DATOS 1

Código ST0245

Insertion Sort



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

After measuring the complexity of both MergeSort and InsertionSort we can conclude that, for a large amount of values, MergeSort is a more effective method. This is because the worst case scenario complexity for InsertionSort is n^2 meanwhile the worst case scenario complexity for MergeSort is $n \cdot \log(n)$.

3.4

Insertion sort isn't an efficient sorting method and has a complexity that can eventually escalate exponentially making it not as efficient for large amounts of data. There are many more efficient methods for sorting data; even merge sort would be more efficient. Since its complexity at its worst case scenario is n^2 then it isn't a recommended method for video games that require fast processing and instant feedback.

3.5

Even though MergeSort has a more effective worst case scenario than InsertionSort, this does not happen for the best case scenario. The complexity for MergeSort will always be $n \cdot \log(n)$ no matter the array it's taking. On the other hand, InsertionSort has a complexity of n in its best case scenario, this case is when the array is already organized. Therefore, when the array is already organized, InsertionSort is much better. Even if the array is organized, MergeSort will still have the complexity of $n \cdot \log(n)$.

3.6 In the coding bat exercise maxSpan we want to find the highest value of a span, which is the number of elements between two numbers inclusive. To create this method, you can define an int variable at 0 from the start. Then create a for loop that traverses the array and an int $j = \text{array.length} - 1$. Then you create a while loop that checks if the value of the array in the for loop is different from the value in array at j . If it is different then $j--$. Then we create another int that is equal to $j - i + 1$. Finally we create a conditional statement that checks if the int created is less than the first int created equal to 0, if it is then equal the variables. Finally return that int.

3.7 The time complexity for the exercises from Array2 in coding bat is $O(n)$. This is because we have simple for loops that only have to be run n times. In this case, n is the length of the array we are dealing with.

3.8 In this case, n is the length of the array we are dealing with.

4) Practice for midterms

4.1 d

4.2 b

4.3 opc

PhD. Mauricio Toro Bermúdez

Professor | School of Engineering | Informatics and Systems

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

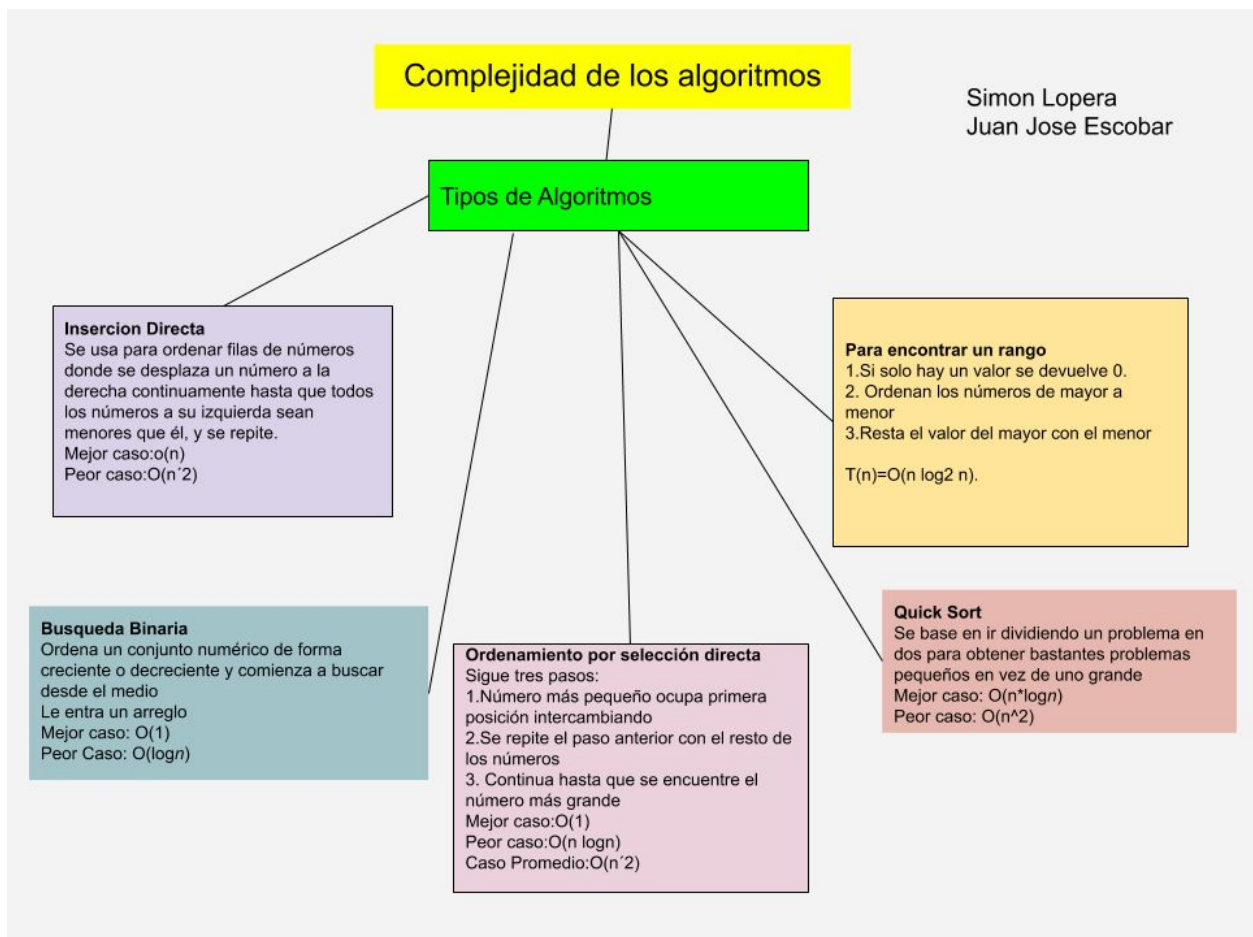
Phone: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS 1

Código ST0245

- 4.4 b
- 4.5 Si termina para todos los números
- 4.6 10000 segundos ya que $c=1/10000$
- 4.7 opc
- 4.8 a
- 4.9 opc
- 4.10 c
- 4.11 c
- 4.12 b
- 4.13 opc
- 4.14 a

5) Recommended reading (optional)



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

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

