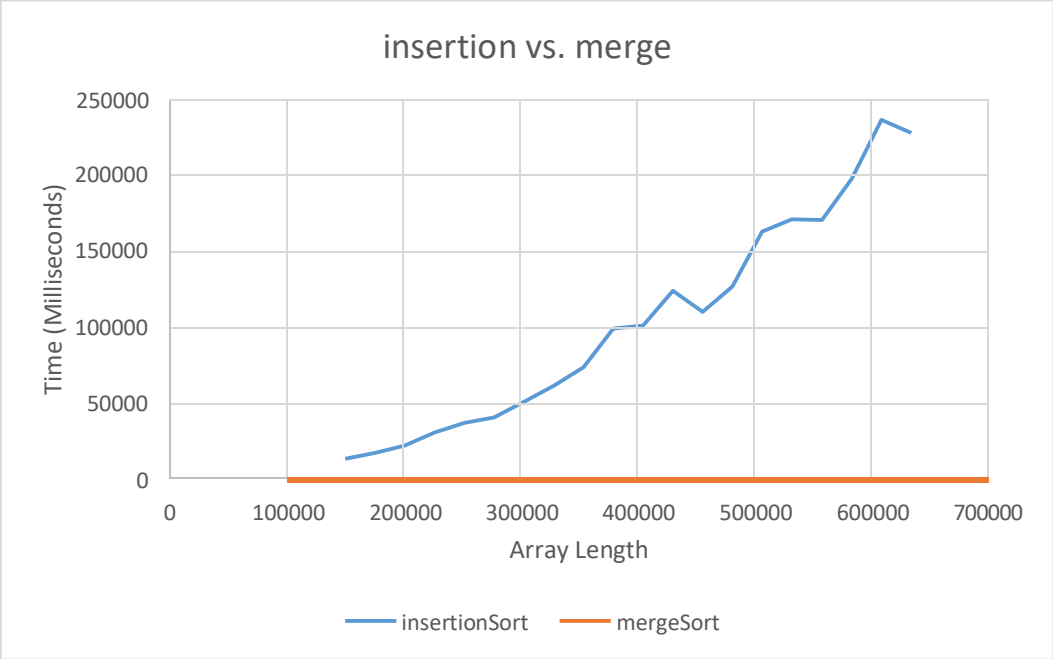
ESTRUCTURA DE DATOS 1
Código ST0245

Laboratory practice No. 2: Algorithm Complexity.

Jose Manuel Fonseca Palacio
Universidad Eafit
Medellín, Colombia
jmfonsecap@eafit.edu.co

Santiago Puerta Florez
Universidad Eafit
Medellín, Colombia
spuertaf@eafit.edu.co

1.1)



2.1) and 2.2) Both of them are solved in the github page.

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

3.1)

Time table for inserionSort()

Array Length	Time (Milliseconds)
100000	4604
101500	5125
103000	5070
104500	5095
106000	5117
107500	5215
109000	5822
110500	6977
112000	6813
113500	5902
115000	6510
116500	6235
118000	6277
119500	7691
121000	8790
122500	9319
124000	8199
125500	7459
127000	7427
128500	8167

Time table for mergeSort()

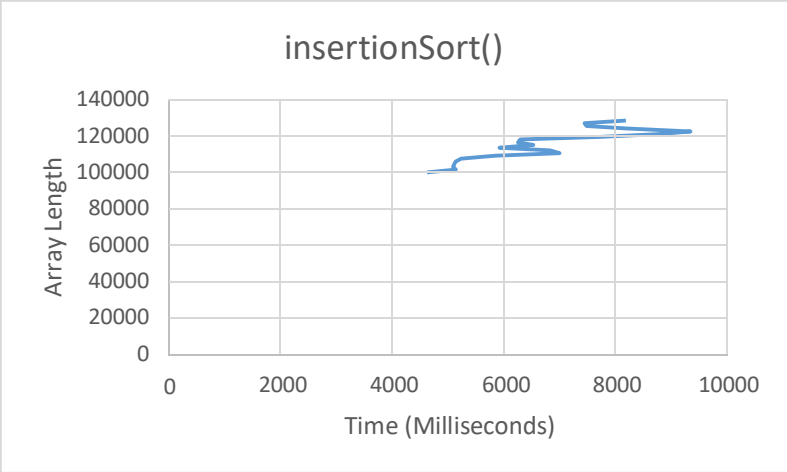
Array Length	Time (Milliseconds)
3551000	6926
3576500	8437
3602000	8116
3627500	8696
3653000	6194
3678500	9706
3704000	6001
3729500	6114
3755000	5372
3780500	8508

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

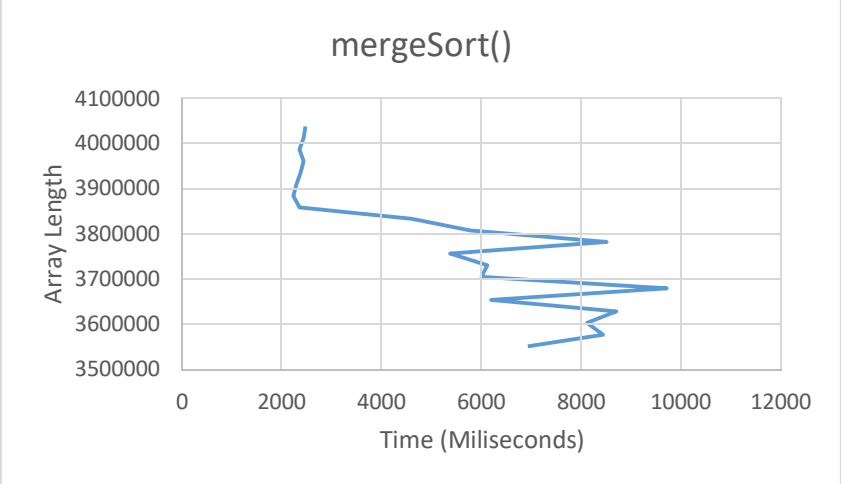
3806000	5772
3831500	4606
3857000	2351
3882500	2227
3908000	2296
3933500	2378
3959000	2433
3984500	2350
4010000	2433
4035500	2475

3.2)

Graph for insertionSort()



Graph for mergeSort()



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

3.3) No, it would not be appropriate to use insertion sort for such a hard job involving tons of data, due to its complexity (O (n^2)). For an-array of 50 million disorganized elements the algorithm would take aproximately 1 hour to sort all elements.

3.4) For the worst case in merge sort appears a log due the function slices the array into different parts and in some cases does not call recursively the function.

3.5) For big arrays if you are wishing insertionSort to be faster than mergeSort the data given to the method insertionSort must be all the same. With all numbers equal insertionSort give us the following table.

insertionSort	mergeSort	
Time (Milliseconds)	Time (Milliseconds)	Array Length
94	1513	4500000
24	2040	6000000
12	3760	7500000
40	7060	9000000

Curiously if you give the mergeSort an already sorted array, the method takes 0 milliseconds to give a response, in contrary with the insertionSort method that takes more time to give an answer than the time taken by the mergeSort. Giving us the following table.

insertionSort	mergeSort	
Time	Time	Array Length
76	0	4500000
14	0	6000000
14	0	7500000
11	0	9000000

3.7)

only14: O(n)
has22: O(n)
evenOdd: O(n)
zeroMax: O(n^2)
sum67: O(n)
seriesUp: O(n^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

3.8) "n" means the quantity of processes that the algorithm has got to make. In some cases, it appears a variable "m" that has a similar function.

4) Practice for midterms

4.1 d
4.2 *b*
4.3 b
4.4 *b*
4.5 .
4.5.1 D
4.5.2 A
4.6 100 segundos
4.7 Todas las anteriores

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

4.8 a
4.9 a
4.10 c.
4.11 c
4.12 b.
4.13 c
4.14 A o 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