ESTRUCTURA DE DATOS 1
Código ST0245

Laboratory practice No. 2: Algorithm complexity

María Alejandra Moncada
Universidad Eafit
Medellín, Colombia
mamoncadaa@eafit.edu.co

Manuela Zapata Mesa
Universidad Eafit
Medellín, Colombia
mzapatam1@eafit.edu.co

1) Project simulation

MERGE SORT	
TIME	N
0.068826914	13300
0.062832355	12300
0.042884827	8500
0.075797558	14300
0.065826654	12800
0.051858187	10200
0.030917883	6400
0.076825619	14700
0.038874865	7600
0.052858353	10600
0.033908844	7100
0.035904169	7400
0.060847759	10500
0.064826965	12700
0.06781888	13100
0.041887283	8200
0.030918121	6500
0.049865961	9900
0.046874046	9500
0.038895845	7900
0.075373173	14400

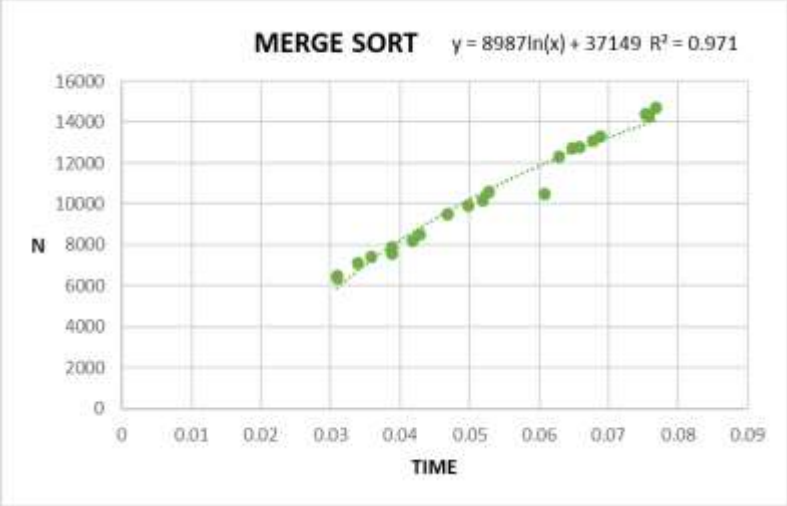
INSERTION SORT	
TIME	N
0.144612074	2100
0.232388496	2700
0.129653215	2000
0.277258396	2900
0.093749285	1700
0.046874523	1200
0.145116806	2100
0.106714249	1800
0.055859804	1300
0.038895845	1100
0.128655672	2000
0.083775997	1600
0.045877218	1200
0.073802471	1500
0.235379934	2700
0.095743179	1700
0.226403475	2600
0.103722572	1800
0.083775759	1600
0.031914949	1000

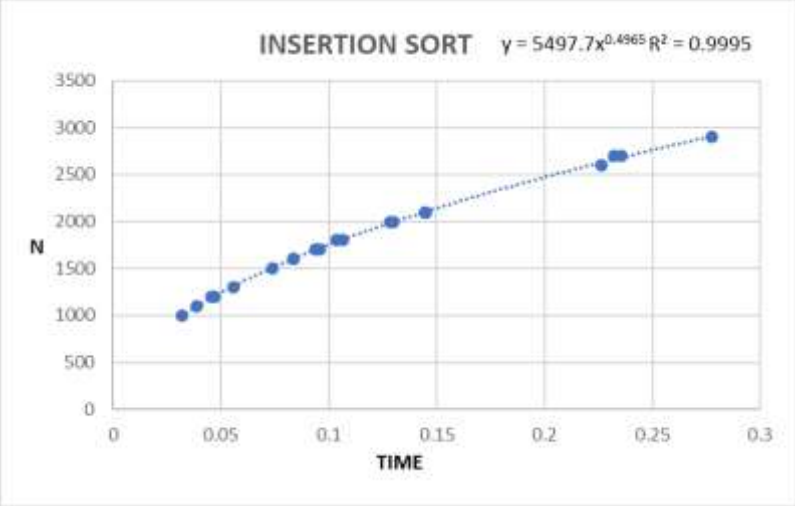
COMPLEXITY MERGE SORT	O(NLOGN)

COMPLEXITY INSERTION SORT	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

UNIVERSIDAD EAFIT
Acreditación Institucional Renovación 2018-2026 Resolución MEN 2158 de 2018
Inspira Crea Transforma | Vigilada Mineducación | www.eafit.edu.co





2) Coding marathon simulation

SquareUp exercise: Elentok, O (2012) CodingBat-Solutions [Source code]
https://github.com/ozelentok/CodingBat-Solutions/blob/master/Java/Array-3.java

Fix34 exercise: Elentok, O (2012) CodingBat-Solutions [Source code]
https://github.com/ozelentok/CodingBat-Solutions/blob/master/Java/Array-3.java

Fix45 exercise: Elentok, O (2012) CodingBat-Solutions [Source code]
https://github.com/ozelentok/CodingBat-Solutions/blob/master/Java/Array-3.java

3) Practice for final project defense presentation

3.1 The table with all data is the same as point 1.
3.2 The graphics requested in this point are the same as the first point.
3.3 We made an average with the times of both Merge Sort = 0.051962554 and Insertion Sort = 0.119008696 and according to this result we can conclude that Merge Sort is more efficient than Insertion Sort by 0.067046142 time units in average.
3.4 Both methods could be efficient to use in a videogame because both are very fast. Insertion Sort would be good to use in a videogame, but it would be better to use Merge Sort.
3.5 For big arrangements the case that will allow Insertion Sort is faster than Merge Sort is when the data of the arrangement are organized, either from highest to lowest or the other way around.

4) Practice for midterms

4.1 *c*
4.2 *b*
4.4 b

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

4.5.1 d
4.5.2 b
4.6 100s
4.7 1 y 2
4.9 d
4.11 c
4.12 b
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