ESTRUCTURA DE DATOS 1
Código ST0245

Laboratory practice No. 2: Algorithm complexity

Santiago Montoya Tobon
Universidad Eafit
Medellín, Colombia
smontoyat@eafit.edu.co

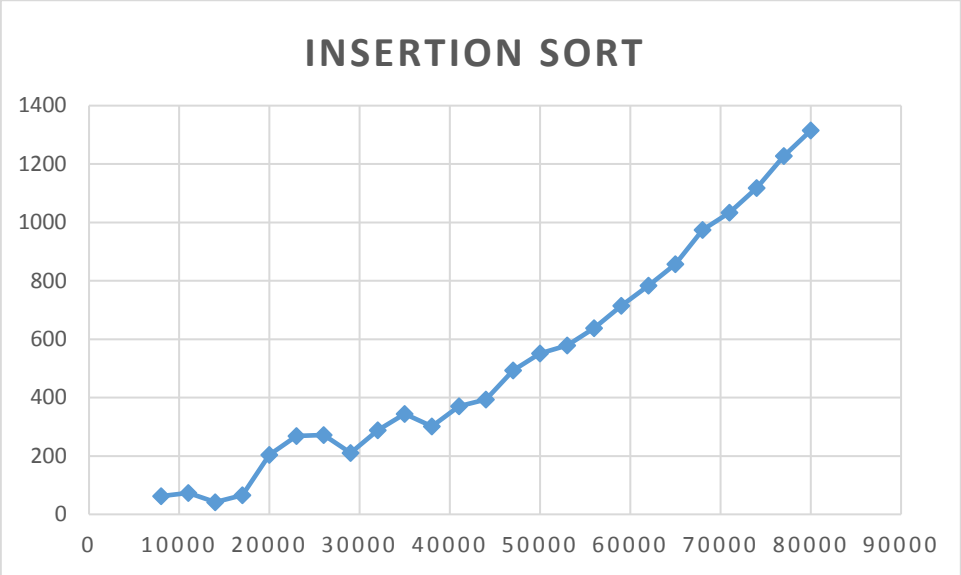
Nelson Andres Barrios Jimenez
Universidad Eafit
Medellín, Colombia
nabarriosj@eafit.edu.co

3) Practice for final project defense presentation

3.1 Insertion sort

Size	Time
8000	63
11000	73
14000	42
17000	66
20000	204
23000	269
26000	272
29000	211
32000	288
35000	344
38000	302
41000	371
44000	394
47000	493
50000	551
53000	579
56000	638
59000	715
62000	784
65000	857
68000	975
71000	1034
74000	1118
77000	1228
80000	1315

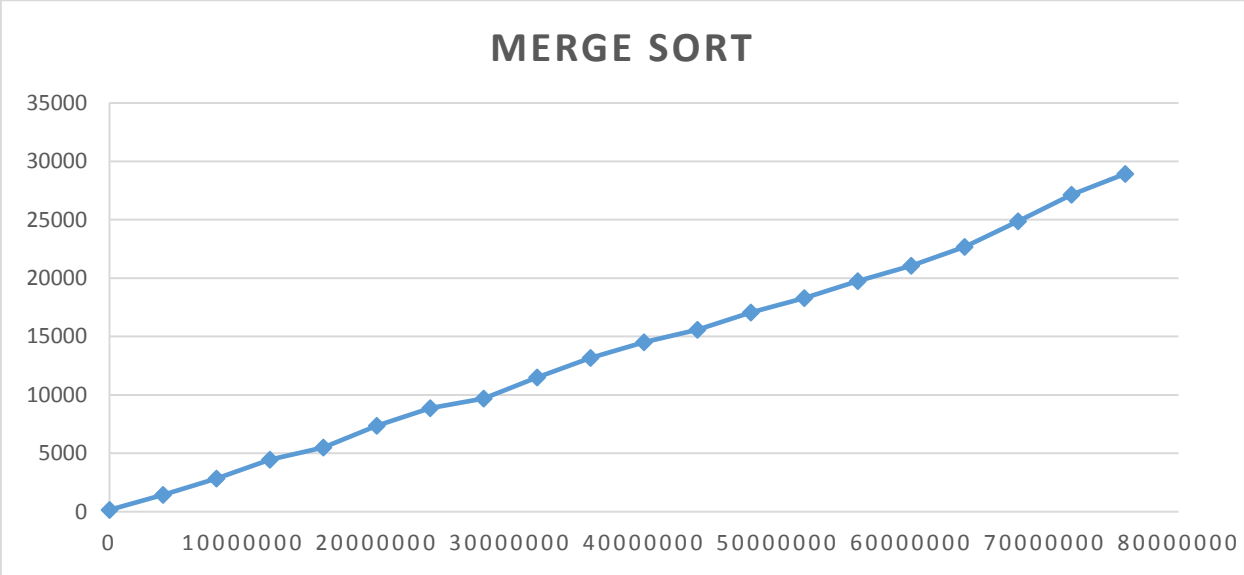
PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473



3.2 Merge sort

Size	Time
10000	31
4010000	1303
8010000	2624
12010000	3849
16010000	5160
20010000	6482
24010000	7825
28010000	9130
32010000	10517
36010000	11838
40010000	13208
44010000	14530
48010000	15921
52010000	17222
56010000	18665
60010000	19980
64010000	21297
68010000	22891
72010000	24962
76010000	27143

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 The Merge Sort algorithm is much more efficient than insertion because Merge sort's complexity is $O(n \times Log(n)$ and Insertion sort´s is On^2. This means that Merge Will take less time to run with very large arrays.

3.4 The Insertion Sort algorithm is not recommended to be use in videogames because the complexity of this is an quadratic equation. This causes the runing time to increase so fast in a case with a large value of n.

3.5 The Insertion Sort algorithm runs quicker than Merge Sort in the cases when most of the elements in the array are already organized.

3.7

-CountEvens: T(n) = C1 +C2n = O(n)
-CenteredAverage: T(n) = C1 + C2n = O(n)
-Sum13: T(n) = C1 + C2n = O(n)
-BigDiff: T(n) = C1 + C2n = O(n)
-has22: T(n) = C1 + C2n = O(n)

3.8 n is the array length

4) Practice for midterms

4.1 O(n+m)
4.2 O(m×n×√n)
4.3 O(ancho)
4.4 O(n^3)
4.5 1) d / 2) a

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

4.6 El algoritmo tomara 100 segundos en ejecutarse.
4.7 1,2,3.
4.8 O(n)
4.9 O(n^3)
4.10 n^2 pasos
4.11 C
4.12 B
4.13
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