Laboratory practice No. 2: Algorithm's Complexity

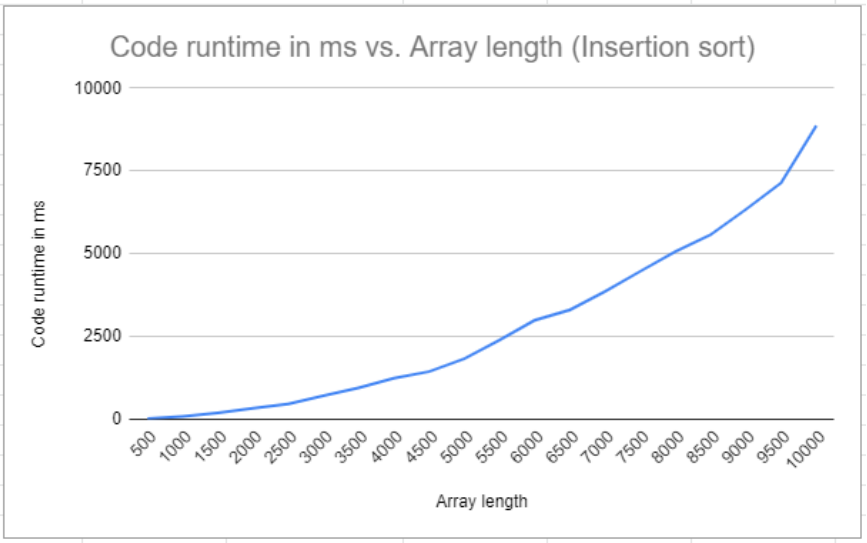
Pedro Botero Aristizabal
Universidad Eafit
Medellín, Colombia
pboteroa@eafit.edu.co

Samuel Ceballos Posada
Universidad Eafit
Medellín, Colombia
sceballosp@eafit.edu.co

3) Practice for final project defense presentation

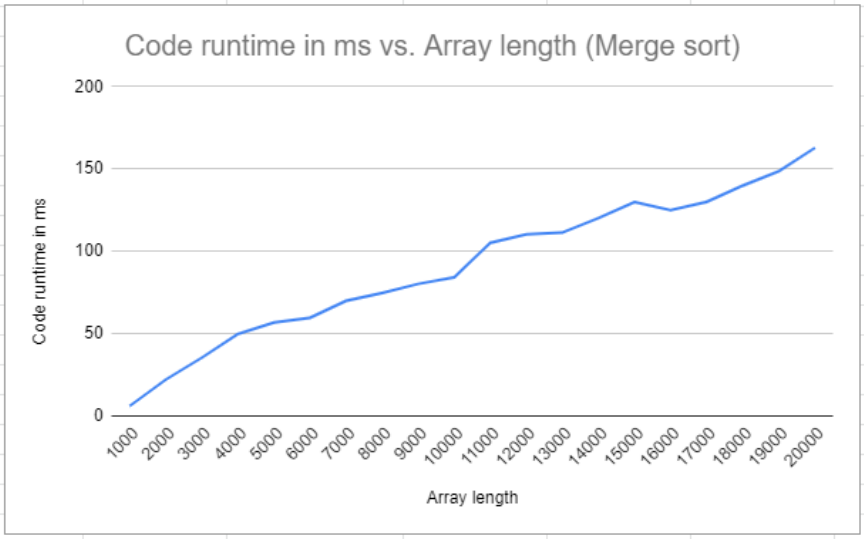
3.1

Insertion Sort	
Array length	Code runtime in ms
500	31.24
1000	93.73
1500	201.89
2000	343.67
2500	466.09
3000	721.12
3500	956.15
4000	1245.16
4500	1445.25
5000	1834.62
5500	2397.93
6000	2994.47
6500	3307.17
7000	3858.96
7500	4471.84
8000	5071.57
8500	5574.63
9000	6337.22
9500	7140.16
10000	8864.59



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

Merge Sort	
Array length	Code runtime in ms
1000	6.1
2000	22.13
3000	35.49
4000	49.83
5000	56.85
6000	59.72
7000	70.04
8000	74.74
9000	80.28
10000	84.28
11000	105.22
12000	110.41
13000	111.35
14000	120.21
15000	129.92
16000	124.99
17000	129.98
18000	139.97
19000	148.6
20000	162.82



3.3 Merge sort is more efficient than insertion sort. If we take a look at the graphs above, we can see that merge sort can sort arrays much faster. For example, it takes insertion sort almost 9 seconds to sort 10,000 numbers, while merge sort can sort 20,000 numbers in only 162.8 milliseconds.

3.4 It wouldn't be ideal to use insertion sort to sort millions of numbers in real time. It would take too much time to render the images, which would make the video game unplayable.

3.5 Insertion sort can be faster than merge sort for big if the most of the numbers in the array are already in order. In this case insertion sort takes less time to run because it has to make less comparisons.

3.6 In our implementation we use two loops to cycle the array twice in order to find out if there is more than one span. The span is calculated by subtracting the position in the array of a number from that same number's last position. If two spans are found, the Math.max function is used to compare both spans in order to see which one is bigger. Finally, the method returns the biggest span.

3.7

Exercise	Level	T(N) Notation	Big O Notation
CountEvens	2	T(n)= n + c	O(n)
BigDiff	2	T(b)= b + c	O(n)
CenteredAverage	2	T(d)= d + c	O(d)
Sum13	2	T(a)= a + c	O(a)
Has22	2	T(f)= f + c	O(f)
MaxSpan	3	T(g)= g^2 + c	O(g^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

Fix34	3	T(h)= h^2 + c	O(h^2)
Fix45	3	T(h)= h + c	O(h)
CanBalance	3	T(i)= i^2 + c	O(i^2)
LinearIn	3	T(j)= k + c	O(k)

3.8

Variable	Explanation
n	The length of an array where the algorithm will count even numbers.
b	The length of an array in which the algorithm will search for the maximum and minimum value
d	The average between the values of an array excluding the maximum and minimum values
a	The sum of an array
f	An array in which the algorithm will check if ther are two consecutive 2
g	The largest span in ana array between two positins with the same number values.
h	Temporal variable that stores a position of the array
i	An array for which you can splt it in two parts so that both sum the same
k	An array for which each value is also located on another given array

4) Practice for midterms

4.1 *b*
4.2 *b*
4.3 b
4.4 b
4.5
- ***4.5.1*** d
- ***4.5.2*** Yes, unless the number is outside the range of type int, which is 2,147,483,648 for both positive and negative ways.

4.6 It would take more or less 100.000 seconds to process 10000 data.
4.7 d
4.8 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

4.9 a
4.10 Optional
4.11 c
4.12 b
4.13 c
4.14 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