Laboratory practice No. 2: Big O Notation

Manuela Zapata Giraldo
Universidad Eafit
Medellín, Colombia
mzapatag1@eafit.edu.co

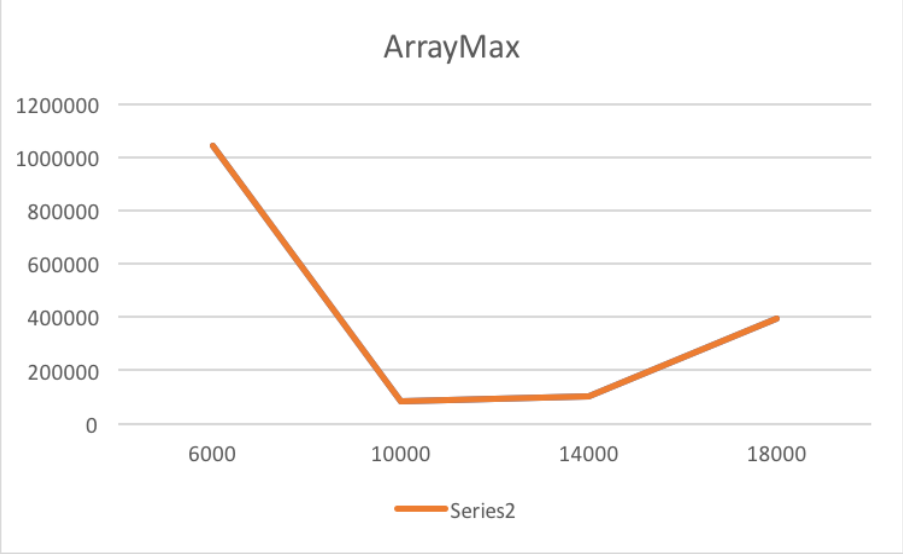
Luis Bernardo Zuluaga
Universidad Eafit
Medellín, Colombia
lbzuluagag@eafit.edu.co

3) Practice for final project defense presentation

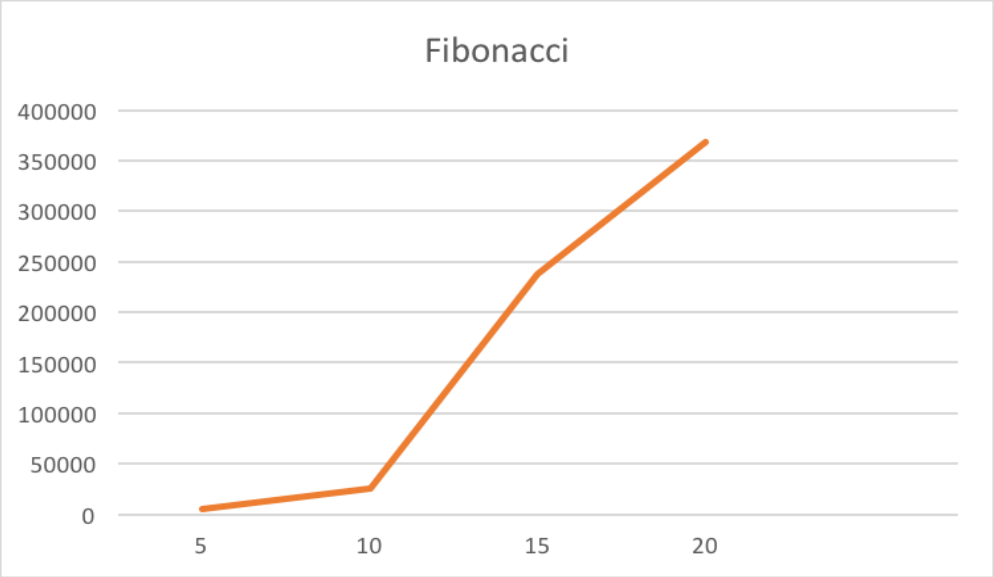
1.

	N = 6000	N = 10000	N = 14000	N = 18000
Array sum	**More than one minute**	**More than one minute**	**More than one minute**	**More than one minute**
Array max	**1044868**	**85404**	**102393**	**395846**
	N = 5	**N = 10**	**N = 15**	**N = 20**
Fibonacci	**5743**	**26191**	**237029**	**368691**

2.



PROFESSOR MAURICIO TORO BERMÚDEZ
Phone: (+57) (4) 261 95 00 Ext. 9473. Office: 19 - 627
E-mail: mtorobe@eafit.edu.co



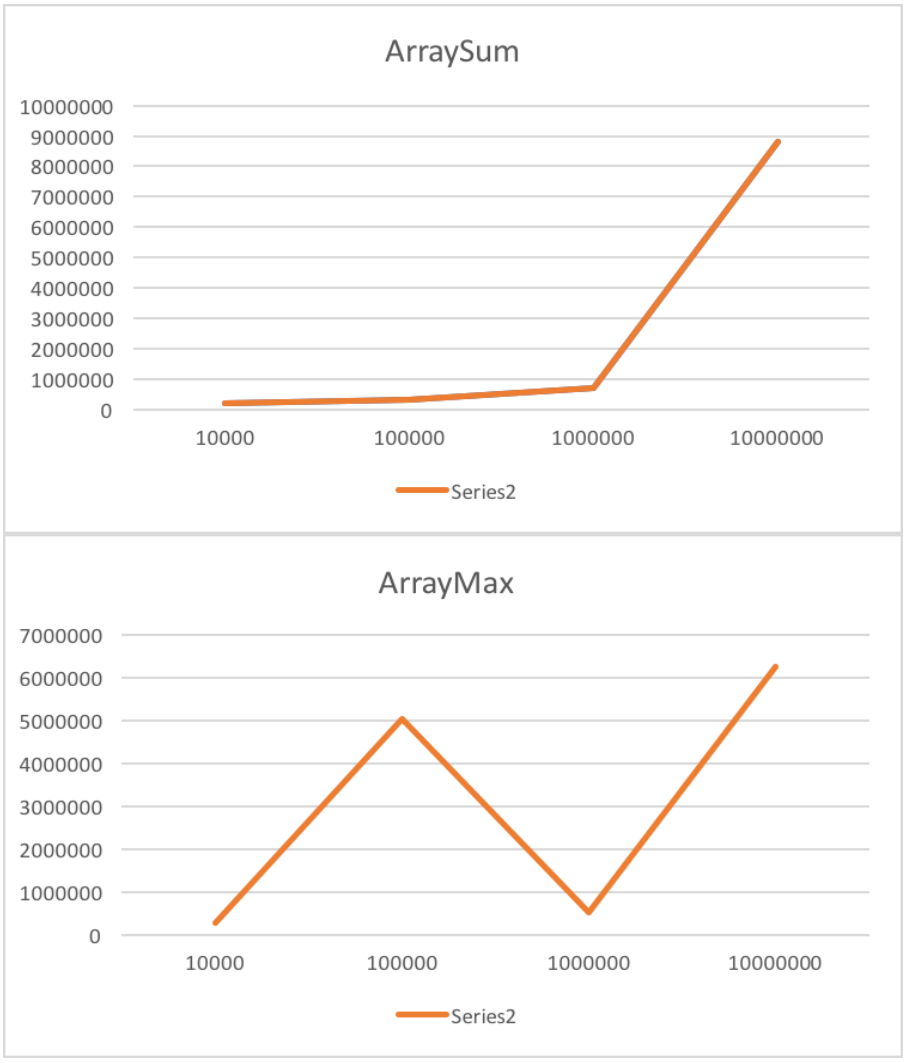
3. Given the experimental data above and the theoretical complexity, we conclude that the graphs do follow the mathematical model of their respective complexity. However, since we used only a handful of data (four values of n) there can be alterations that make the graphs look a little out of shape. But their overall shape does fit into the mathematical model of their complexity.

4.

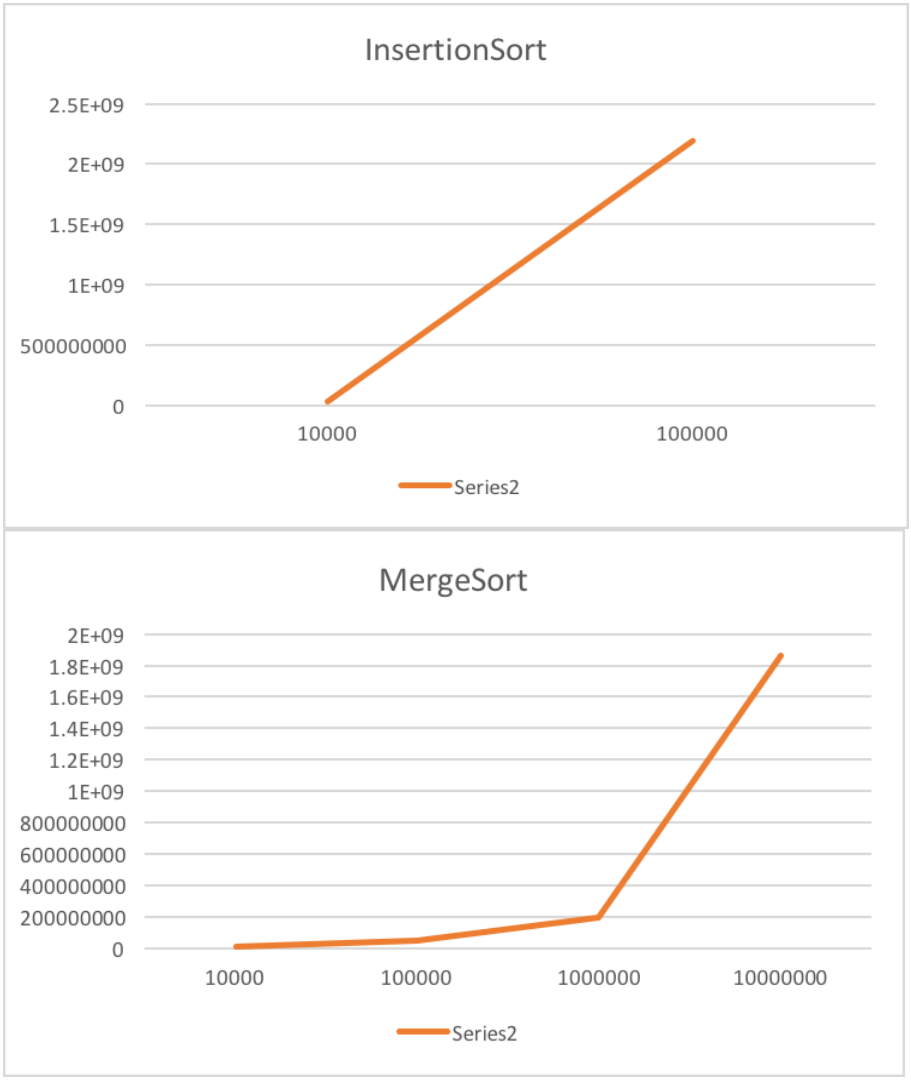
	N = 10.000	N = 100.000	N = 1'000.000	N = 10'000.000
Array sum	**190858**	**322614**	**682092**	**8801654**
Array max	**281835**	**5019117**	**516946**	**6235001**
Insertion sort	**34135064**	**2190836413**	**More than one minute**	**More than one minute**
Merge sort	**6331683**	**45672986**	**195207975**	**1862147417**

5.

PROFESSOR MAURICIO TORO BERMÚDEZ
Phone: (+57) (4) 261 95 00 Ext. 9473. Office: 19 - 627
E-mail: mtorobe@eafit.edu.co



PROFESSOR MAURICIO TORO BERMÚDEZ
Phone: (+57) (4) 261 95 00 Ext. 9473. Office: 19 - 627
E-mail: mtorobe@eafit.edu.co



6. Given the experimental data above and the theoretical complexity, the graphs do not meet the mathematical model since we have few data (four values of n) and it is not possible to make out the ideal shape that the graph should have. Additionally, we can tell that the algorithm for InserionSort is not very efficient considering that we were only able to get data for two values of n. The program took too long to process for remaining two values.
7. In InsertionSort, for bigger n values, the algorithm does not run or takes a very long time to process. It's complexity is O(n^2).
8. Unlike InsertionSort, with bigger values of n, ArraySum is very efficient. It's complexity is O(n), which means that no matter how big the input data gets, the algorithm is able to process it fast and efficiently without much difference in time in between.
9. MergeSort is much more efficient than InsertionSort, considering that it's complexity is O(n(log n)). We can also see this in the graphs above. The algorithm for MergeSort was able to successfully process all input data, whereas the algorithm for InsertionSort was only able to process two.
10. .

PROFESSOR MAURICIO TORO BERMÚDEZ
Phone: (+57) (4) 261 95 00 Ext. 9473. Office: 19 - 627
E-mail: mtorobe@eafit.edu.co

11. Codingbat Array2

- Sum13:
  T(n) = c1+c2+c3+c4+n(c5+c6+c7+c8)+c9
  T(n)=O(n)
- Sum67:
  T(n) = c1+c2+n(c3+c4+c5+c6+c7+c8+c9+c10+c11)+c12
  T(n)=O(n)
- Has22:
  T(n) = c0+n(c1+c2+c3+c4)+c5
  T(n)=O(n)
- Luky13:
  T(n) = c0+n(c1+c2+c3)+c4
  T(n)=O(n)
- Sum28:
  T(n) = c0+c1+n(c2+c3+c4)+c5
  T(n)=O(n)

Codingbat Array3

- CanBalance:
  T(n) = c0+c1+c2+n(c3+c4+c5+10+11+12)+nm(c6+c7+c8+c9)+c13
  T(n)=O(nm)
- SeriesUp:
  T(n)=c0+c1+c2+c3*n+nm(c4+c5+c6)+c7
  T(n)=O(nm)
- Fix45:
  T(n)= c+c1+n(c2+c3+c4+c5+c6+c7)+c8
  T(n)=O(n)
- Fix34:
  T(n) = c0+c1+n(c2+c3+c4+c5)+nm(c6+c7+c8)+c9
  T(n)=O(nm)
- SquareUp:
  T(n) = c0+c1+c2+c3+c4+n(c5+c6+c10)+nm(c7+c8+c9)+c11
  T(n)=O(nm)

4) Practice for midterms

1. *c*
2. *b*
3. *b*
4. *b*
5. d
6. d
7. .1 T(n)=(n-1)+c

7.2 O(n)

PROFESSOR MAURICIO TORO BERMÚDEZ
Phone: (+57) (4) 261 95 00 Ext. 9473. Office: 19 - 627
E-mail: mtorobe@eafit.edu.co