# **Laboratory practice No. 2:**

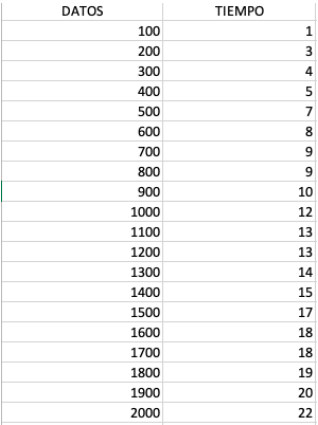
# **Algorithm Complexity**

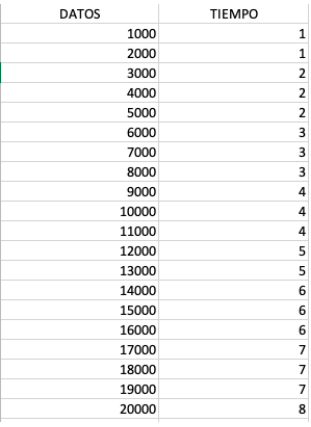
|  |  |  |
| --- | --- | --- |
| **Juan Pablo Madrid Florez**  Universidad Eafit  Medellín, Colombia  jpmadrid@eafit.edu.co | **Abelino Sepúlveda Estrada**  Universidad Eafit  Medellín, Colombia  asepulvede@eafit.edu.co | **Andrés Gómez Arango**  Universidad Eafit  Medellín, Colombia  Agomeza10@eafit.edu.co |

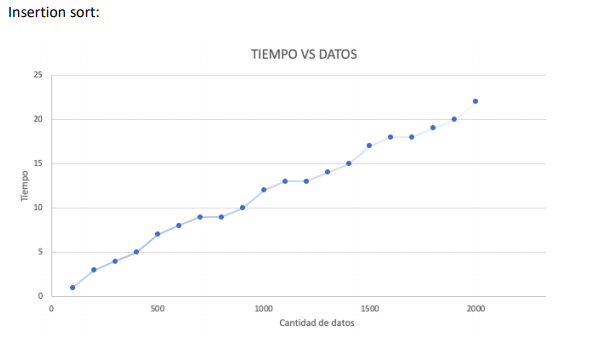
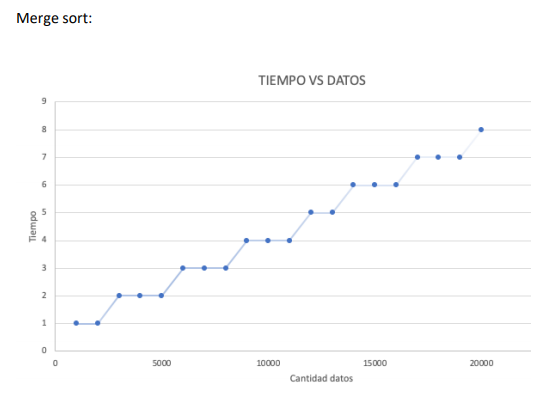
**3) Practice for final project defense presentation**

**3.1**

Insertion Sort Marge Sort





**3.2**  

**3.3**

As can be seen in the graphs and tables, marge sort is much more efficient with respect to longer arrays, since they execute them in a much shorter time than the insertion sort, as example the length 1000 in Insertion took 12 seconds and marge 1 second, there we can see the big difference

**3.4**   
We believe that it is not appropriate since insertion sort is too slow with arrays that do not exceed 10,000 elements, so, having an array with millions of elements, insertion sort would be obsolete and would make the game very slow

**3.5**

for insertion sort to be faster, the data must be in a more organized way to be more effective than marge sort

**3.7 y 3.8**

Array 2:

countEvens:T(n)= T(n-1)+c, que es O(n) (n is the array´s length)

bigDiff: T(n)= T(n-1)+c, que es O(n) (n is the array´s length)

centeredAverage: T(n)= T(n-1)+c, que es O(n) (n is the array´s length)

sum13: T(n)= T(n-1)+c, que es O(n) (n is the array´s length)

sum67: T(n)= 2T(n-1)+c, que es O(n^2) (n is the array´s length)

Array 3:

maxSpan: T(n)= 2T(n-1)+c, que es O(n^2) (n is the array´s length)

fix34: T(n)= 2T(n-1)+c, que es O(n^2) (n is the array´s length)

fix45: T(n)= 2T(n-1)+c, que es O(n^2) (n is the array´s length)

canBalance: CountEvens T(n)= T(n-1)+c, que es O(n) (n is the array´s length)

linearIn: T(n)= 2T(n-1)+c, que es O(n^2) (n is the array´s length)

***4) Practice for midterms***

* 1. *c*
  2. b
  3. (opcional)
  4. b
  5. d

4.5.2 a

* 1. 10 s
  2. The propositions 1, 2 and 3 are true
  3. (opcional)
  4. a
  5. (opcional)
  6. c
  7. b
  8. (opcional)
  9. a