

Laboratorio Nro. 2: Notación O

Juan Pablo Castaño Duque

Universidad Eafit

Medellín, Colombia

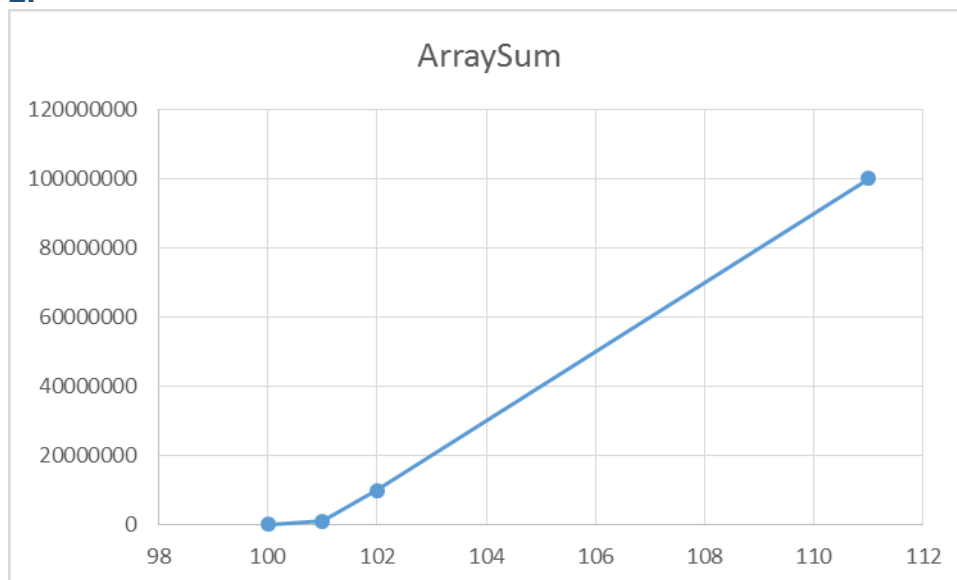
jpcastanod@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

1.

	N= 100.000	N= 1'000.000	N= 10'000.000	N= 100'000.000
ArraySum	100	101	102	111
ArrayMax	103	101	118	123
InsertionSort	192	8173	+ de 5 mn	+ de 5 mn
MergeSort	102	114	293	2462

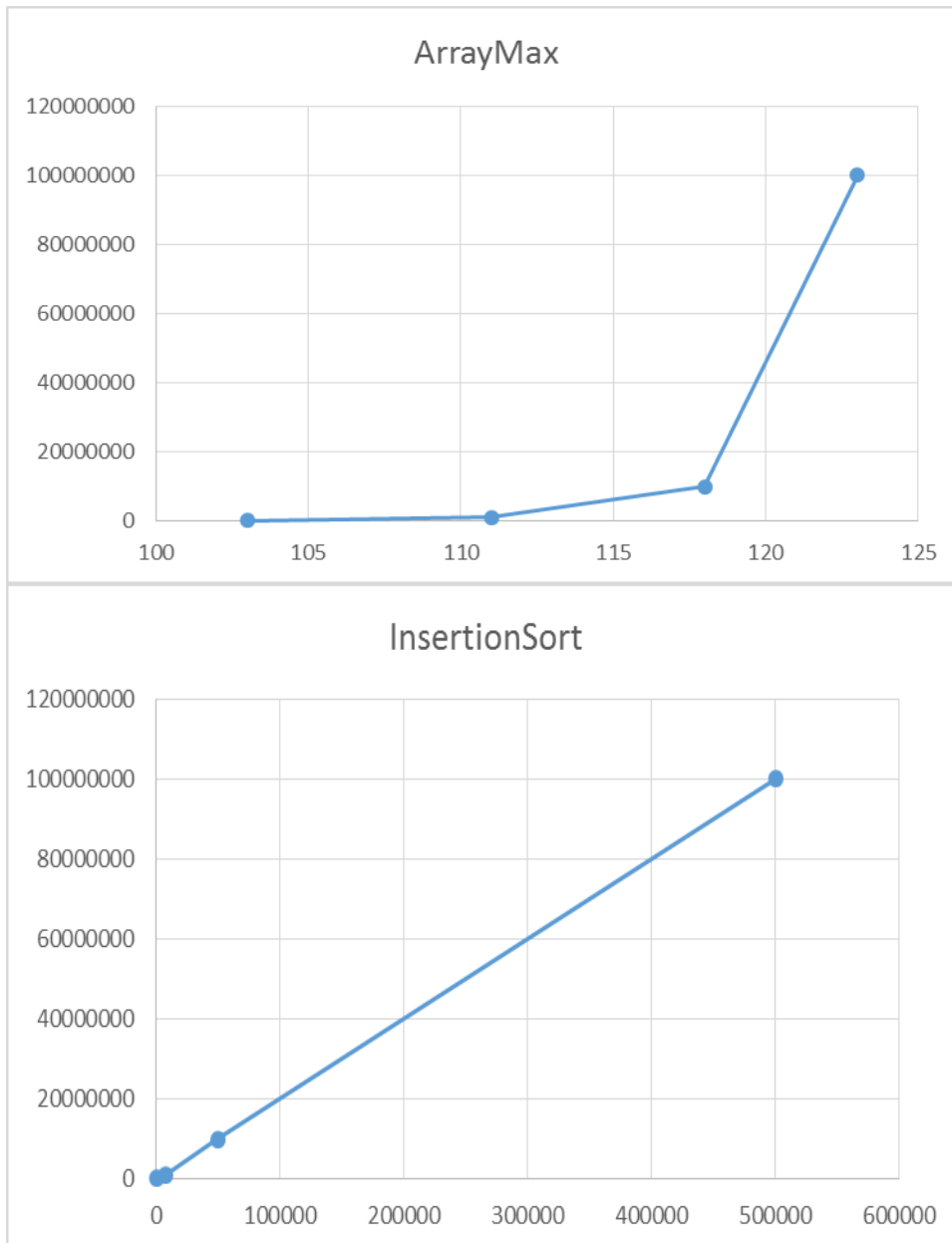
2.

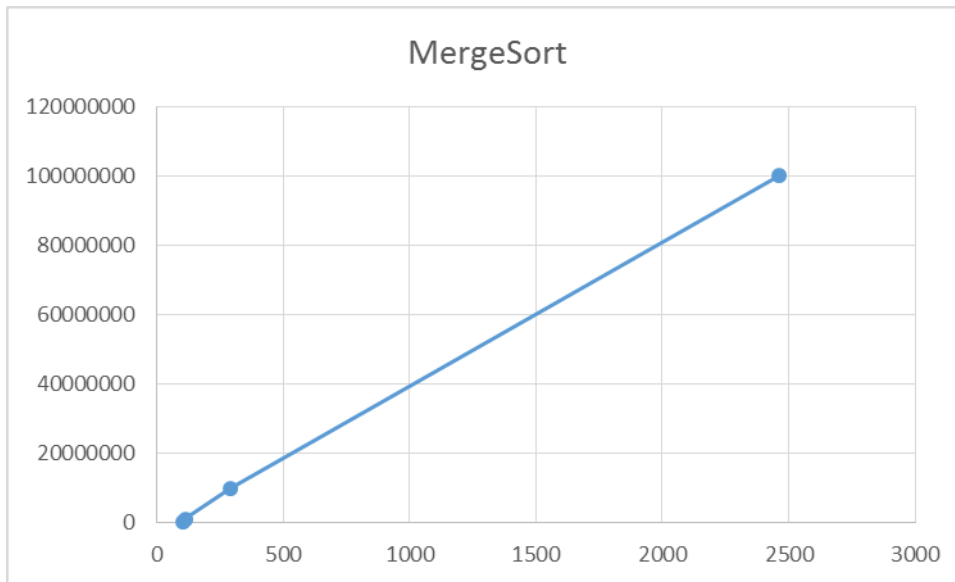


DOCENTE MAURICIO TORO BERMÚDEZ

Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627

Correo: mtorobe@eafit.edu.co





3. Se concluye que entre menor sea la complejidad del algoritmo, menor tiempo toma en realizar operaciones muy grandes.
4. Debido a que la complejidad de InsertionSort es muy grande a comparación de las demás, a medida que se va haciendo mas grande N, esto va tomando valores de tiempo muy altos.
5. En el método ArraySum sucede que el tiempo es casi que constante, debido a que la complejidad de este algoritmo es $O(n)$, la cual es muy pequeña a comparación del metodo InsertionSort.
6. Para arreglos muy grandes, mergesort es muchisimo mas efectivo, reduce el tiempo notablemente a comparación de InsertionSort, en cambio para arreglos muy pequeños, los dos trabajan casi que a la para, debido a que en procesos de pequeñas cantidades, la complejidad no se ve tan diferente en cuanto al tiempo de ejecución.
7. Max es un método que recibe un arreglo de enteros, con este arreglo coge la aparición mas a la izquierda y mas a la derecha de un número del arreglo y cuenta la cantidad de números que hay entre ambos incluyendolos.

8. CountEvents:

- ☐ $O(n \cdot C + C)$
- ☐ $O(n \cdot C)$
- ☐ $O(n)$

BigDiff:

- ☐ $O(n \cdot C + C)$
- ☐ $O(n \cdot C)$
- ☐ $O(n)$

No14:

- ☐ $O(n \cdot C + C)$
- ☐ $O(n \cdot C)$
- ☐ $O(n)$

Sum13:

- ☐ $O(n \cdot C + C)$
- ☐ $O(n \cdot C)$
- ☐ $O(n)$

More14:

- ☐ $O(n \cdot C + C)$
- ☐ $O(n \cdot C)$
- ☐ $O(n)$

Fix34:

- ☐ $O(n^2 + n \cdot c + C)$
- ☐ $O(n^2 + n \cdot C)$
- ☐ $O(n^2 + n)$
- ☐ $O(n^2)$

LinearIn:

- ☐ $O(n \cdot C + C)$
- ☐ $O(n \cdot C)$
- ☐ $O(n)$

maxSpan:

- ☐ $O(n^2+n*c+C)$
- ☐ $O(n^2+n*C)$
- ☐ $O(n^2+n)$
- ☐ $O(n^{2*})$

CanBalance:

- ☐ $O(n^2*m+n*c+C)$
- ☐ $O(n^2+m*n*C)$
- ☐ $O(n^2+m*n)$
- ☐ $O(n^2)$

Fix45:

- ☐ $O(n^2+n*c+C)$
- ☐ $O(n^2+n*C)$
- ☐ $O(n^2+n)$
- ☐ $O(n^2)$

9. C:

Es la suma de varias constants, es decir uno o varios procesos que se repite un número ya determinado de veces.

n:

Es una acción que se repetirá tantas veces como se deba según el proceso que se desee realizar y esta va a variar según la cantidad de datos que sean ingresados.

m:

Cumple exactamente el mismo papel de n, pero m se utiliza cuando hay dos procesos de este en el mismo algoritmo, pero con diferente cantidad de repeticiones, por ejemplo en dos ciclos diferente ya que si estos fueran iguales se utilizaría n^2

4) Simulacro de Parcial

1. *c*
2. *d*
3. *b*
4. *b*
5. *d*