

# Worshop 3: Closest Pair\*

Daniel Olivares  
Universidad del Norte  
Barranquilla, Colombia  
daolivares@uninorte.edu.co

**Resumen**—The purpose of this article is to show the performance and functioning of an algorithm that is in charge of generating a file with random numbers and then counting how many times these are repeated this was made by experiments using a specialized software in which the algorithm mentioned was implemented and tested multiple times.

## INTRODUCCIÓN

En la actualidad es importante crear algoritmos que resuelvan problemas con mayor eficiencia. Como ingenieros de sistemas el analisis de como estos algoritmos funcionan y cual es su tiempo de respuesta es imprescindible para la optimizacion de los mismos. Es por eso que el analisis de algoritmos como el de *Closest Pair* se hace necesario y fundamental hacerlo.

## DEFINICION DEL PROBLEMA

Un algoritmo es un “conjunto ordenado y finito de operaciones que permite hallar la solución de un problema”, al implementarlos hacer un analisis detallados del comportamiento del mismo es indispensable para conocer el tiempo de complejidad y los casos que presenta y saber si el algoritmo es eficiente. El algoritmo *ClosestPair* al ser tan importante por sus aplicaciones se debe analizar detalladamente.

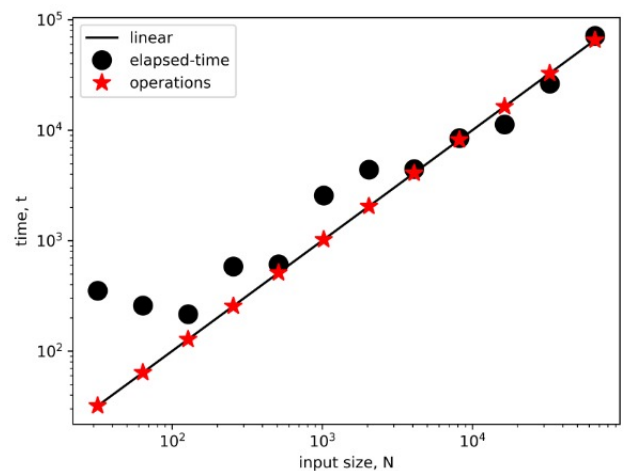
## METODOLOGIA

El algoritmo se realizó por medio de un software especializado llamado *Java Netbeans*. Las graficas fueron hechas en *Visual Studio Code* usando *Python*.

Se tuvieron en cuenta, diferentes tamaños y repeticiones por los mismo, para mas precision. El codigo completo se puede encontrar en mi repositorio de GitHub <https://github.com/daolivares/ClosestPair>

## RESULTADOS

Los resultados obtenidos son los siguientes:



## DISCUSIÓN

Como podemos ver el algoritmo cumple con la teoria dada en clases y analizada antes, este tipo de algoritmo como *DivideAndConquer* tiene un time complexity de  $N$ , el cual es mucho mas eficiente que el logaritmo *Brutal-Force*, el cual, como se sabe por las experimentaciones que hizo el profesor Misael, es cuadratico. Ahora, es importante analizar el porque algunos valores no cumplen con la linealidad. Podria ser la falta de repeticiones o que en valores pequeños de  $N$ , ocurren otros procesos en la computadora.

## CONCLUSIONES

Se pudo reafirmar y demostrar la teoria vista en clases, el algoritmo si cumple con un time complexity de  $N$ , lo cual facilitara en tiempo y recursos, la resolucion del problema en el cual se aplica. Además, concluyo que este tipo de analisis ayudan a mejorar los algoritmos al conducir una prueba real aplicada.

## REFERENCIAS

1. **Time Elapsed:**  
<https://stackify.com/heres-how-to-calculate-elapsed-time-in-java/>
2. **Graphic creator code:** . Diaz  
Maldonado, M. `loglogplot.py`.  
<https://github.com/misael-diaz/computer-programming/blob/main/src/io/java/loglogPlot.py> (2022).